Dynamic Modeling of a Mobile Air Conditioning System From Actual Performance Data

C. R. Schenk, L. A. Knobloch and R. R. Crawford

ACRC TR-10

January 1992

For additional information:

Air Conditioning and Refrigeration Center
University of Illinois
Mechanical & Industrial Engineering Dept.
1206 West Green Street
Urbana, IL 61801

(217) 333-3115

Prepared as part of ACRC Project 09
Mobile Air Conditioning Systems
R. R. Crawford, Principal Investigator

The Air Conditioning and Refrigeration Center was founded in 1988 with a grant from the estate of Richard W. Kritzer, the founder of Peerless of America Inc. A State of Illinois Technology Challenge Grant helped build the laboratory facilities. The ACRC receives continuing support from the Richard W. Kritzer Endowment and the National Science Foundation. The following organizations have also become sponsors of the Center.

Acustar Division of Chrysler
Allied-Signal, Inc.
Amana Refrigeration, Inc.
Bergstrom Manufacturing Co.
Caterpillar, Inc.
E. I. du Pont de Nemours & Co.
Electric Power Research Institute
Ford Motor Company
General Electric Company
Harrison Division of GM
ICI Americas, Inc.
Johnson Controls, Inc.
Modine Manufacturing Co.
Peerless of America, Inc.
Environmental Protection Agency
U. S. Army CERL
Whirlpool Corporation

For additional information:

Air Conditioning & Refrigeration Center
Mechanical & Industrial Engineering Dept.
University of Illinois
1206 West Green Street
Urbana IL 61801

217 333 3115

TABLE OF CONTENTS

Chapter		Page
1	INTRODUCTION	1
	Objectives	1
	Background	2
2	LITERATURE REVIEW	3
	Introduction	3
	Steady-State Modeling Techniques	3
	Transient Modeling Techniques	5
	Implications of the Different Modeling Techniques	7
3	MODEL DEVELOPMENT	8
	Introduction	8
	Test Facility	8
	Model Development Procedure	10
	System Model Developer	14
	Data Reduction Program	16
4	SYSTEM MODEL	17
	Test Plan	17
	Compressor Speed	22
	Inlet Air Temperature	26
	Condenser Air Flow	30
5	CONCLUSION	35
	Summary	35
	Recommendations	37
REFERENCES		39
APPENDIX A		41
APPENDIX B		61

Chapter 1

INTRODUCTION

Objectives

In the past decade, the automotive industry has been under increasing pressure to improve fuel economy and performance. To achieve this improvement, there has been a movement toward smaller, lighter cars. As the cars become smaller, the power and space requirements of the air conditioning system become more critical. There is therefore a need to minimize the size, weight and power of the air conditioning system. To help facilitate these goals, advanced control algorithms may be used. These algorithms could provide the optimum control settings for the expansion valve, compressor displacement, or airflow rates for any given set of conditions. It would be desirable to have an advanced control algorithm compact in both the memory and the amount of processor time required. These advanced control algorithms need to be able to predict the system behavior to changes in the inputs. The objective of this study is to provide a compact, low order system model that may be utilized in an optimal or adaptive control algorithm.

Mobile air conditioning systems almost always operate under transient conditions. The inside and outside temperatures, airflow over the condenser and evaporator, and engine speed all are subject to frequent changes. Depending on how the control algorithm is configured, the compressor may also cycle on and off. For these reasons the system model must be able to predict the transient behavior of the system.

The objective of this study will be to develop a transient system model for advanced control algorithms. The significant independent variables and their coefficients will be determined for the compressor power, evaporator air outlet temperature, and evaporator and condenser refrigerant pressures. These coefficients will be derived from data collected from a mobile air conditioning test stand. This facility is described in detail (Knobloch 1992). The test procedure is also described.

Background

The mobile air conditioning empirical modeling study is part of the research being conducted at the Air Conditioning and Refrigeration Center (ACRC) at the University of Illinois at Urbana-Champaign.

The original test facility was constructed by Michael (1989). The facilities have been considerably expanded for this study. The system has been fully instrumented, measuring refrigerant pressures, temperatures, and flow rates at the inlet and outlet of each of the components as well as the airflow rates and temperatures. The means to manually control the airflow rates, temperatures, and the expansion valve have also been added.

Chapter 2

LITERATURE REVIEW

Introduction

There has been a great deal of research done on steady state and transient vapor-compression system modeling. The literature is investigated to provide a background knowledge of the fundamental system equations, transient solution methods, and experimental techniques employed in previous studies. Most of these models incorporate varying degrees of physical and empirical parameters. The goal of this study is to develop a compact automotive air conditioning system model. Since little research in this area is available, these papers will provide insight into the techniques that may be used for this study.

The literature review is divided into three sections. The first is a review of the steady state vapor-compression system modeling techniques. The second section deals with the development of various transient system models. These models are roughly in the order of decreasing reliance on fundamental principles, i.e., increasingly empirical in nature. The third section discusses the implications of the various modeling techniques.

Steady-State Modeling Techniques

Davis and Scott (1976) proposed a steady-state system model that combines empirical parameters with physical relationships. The modeling algorithm was divided into

component modules which were iterated until the flow rates and superheats converged. Modules were developed for the compressor, condenser, expansion valve, and evaporator. Fundamental relationships such as mass and energy balances were utilized for most of the operating characteristics of the components. The empirical values consisted of loss terms and efficiencies which were found from performance data or estimated. This simulation method proved to be considerable faster than purely analytical methods. The accuracy of 5% was also found to be acceptable for systems design work.

A similar modular program was developed by Dhar and Soedel (1979) as a system design tool. The program is combination of fundamental engineering principles and empirical parameters such as heat transfer coefficients, oil transportation, etc. These parameters may be found from tests on the components. The mathematical model consists of ordinary differential equations and algebraic equations. The equations were solved numerically, using the Euler method on the differential equations. A fourth-order Runge-Kutta method was considered, but was found to be much more time consuming than the Euler method for a given degree of accuracy. The model was used to predict the transient behavior of a window air conditioner and a heat pump.

A steady-state simulation of an automotive air conditioning system was developed by Davis, Chianese and Scott (1972). The models for each of the components were derived from physical and empirical quantities. The evaporator and condenser were modeled using the ε-NTU method. The balance point for each of the components was found by iteration of the component equations. The system balance point was found by iterating through the evaporator, compressor, and condenser, respectively, until the condenser capacity converged. A model of the vehicle passenger compartment was also developed and integrated with the air conditioning simulation program. The simulation results were very close to those found by experimentation.

Cecchini and Marchal (1991) also used a combination of fundamental and empirical terms in their system model, with the goal of minimizing the amount of testing required to determine empirical parameters. To achieve this, steady state operation was assumed. With only two or three testing points for each component, the performance uncertainty of three water-to-water heat pumps was ±5%. For air conditioners of various capacities, the performance uncertainty was ±10%. Frosting/defrosting operations were also tested, resulting in an uncertainty of ±10%.

Transient Modeling Techniques

A vapor-compression heat pump model based entirely on first principles was developed by MacArthur (1984). The model was internally verified in that the conservation laws were satisfied, and therefore no attempt was made to validate it against specific heat pump data. The equations for each of the system components and the numerical solution techniques necessary were presented. Search methods and iterative procedures were used to yield the solutions. This model was intended as a research tool for the development of more effective heat pump controls.

Start-up transients for a vapor-compression refrigerating system were modeled by Josiassen (1978). Mass distribution during start-up was emphasized in developing the system model. Mass "pots" were used to describe the compressor housing, condenser, and evaporator. The "pot" was defined as: a) a volume in which refrigerant was present at stagnation conditions, b) the pot shell had the same temperature as the refrigerant, c) thermal capacity, but no resistance in the shell, and d) loss-free inlet and outlet tubes. The pressure drops were ignored to simplify the calculations. The integration of mass, pressure

and temperature were evaluated using the Euler method. The trends determined using the model were similar to the experimental trends.

Mitsui (1988) also modeled the start-up transients. He developed the model from the fundamental differential equations and solved them using the Euler's method. This model correctly predicted the experimental trends. The model was used to investigate a PID-controlled expansion valve. The controller used the discharge superheat and the compressor speed as inputs, with the compressor speed used in a feedforward mode to compensate for the lag in the superheat measurement. The electronically controlled valve showed an increase in system efficiency during the start-up phase and allowed a 3°C lower evaporator discharge air temperature.

Crawford and Woods (1985) developed a method for deriving a dynamic system model from actual performance data. The method used an autoregressive least-squares technique to determine the model coefficients. The indoor dry-bulb and mean radiant temperatures of a single-family residence were calculated as a function of ambient temperature, solar radiation, and heat input. The indoor dry-bulb temperature (T_{db}) was modeled using a ΔT_{db} model form, where ΔT_{db} is the difference between T_{db} at time k and k-1. The number of parameters was reduced by using a steady-state rationalization of the indoor and outdoor temperatures. Seven terms were found to be significant, and gave a standard deviation of model error of 0.164°C. The mean radiant temperature was modeled using a similar form. A four-term model resulted in a standard deviation of model error of 0.203°C.

The same method was utilized by Shirey (1987) to model a residential heat pump. This model differed from that of Crawford and Woods (1985) in that the model had to be able to predict the effect of outdoor temperature on the performance of the heat pump. The

indoor and outdoor temperatures, solar radiation, wind, and heat pump power consumption were recorded. Data were taken for heating and cooling conditions and models were developed for each situation. The heating model used a ΔT form with nine parameters to achieve a standard model deviation of 0.15°F. The cooling model used the difference between the indoor and outdoor temperatures as the dependent variable. The model consisted of eight parameters and gave a standard deviation of 0.19°F.

Implications of the Different Modeling Techniques

Since the goal of this study is to develop a compact model of an automotive air conditioning system which is subject to many transients, the steady-state modeling techniques were not directly applicable. The transient methods which incorporated first principles required the solution of differential equations. This was achieved by using the Euler and Runge-Kutta integration methods, both of which were used or considered by Josiassen and Mitsui. Both of these methods are computationally cumbersome, so the entirely empirical method presented by Crawford and Woods was chosen for the model development.

Chapter 3

MODEL DEVELOPMENT

Introduction

The system modeling procedure and model development program are described in this chapter. The auto-regressive least-squares algorithm utilized for the modeling process is detailed. This algorithm is implemented by the "System Model Developer" program. A brief description of the program and its capabilities is presented. This program is used in the overall system model development and the component modeling (Knobloch 1992). The program used to analyze the raw data is also presented.

Test Facility

The test stand (Michael 1989, Knobloch 1992) was built around the air conditioning system of a 1989 Chrysler K-car. This system (Figure 3.1) was composed of a swash-plate compressor, a Skyve-Fin™ condenser, and a plate-fin evaporator. The system originally had a thermal expansion valve which maintained the evaporator superheat at about 20°F. This valve was modified to allow the superheat to be adjusted manually, since the unmodified valve had a tendency to oscillate. These pressure oscillations could not be easily modeled.

For this study, nine system parameters were measured and recorded. Of these, five of the parameters were on the air-side. The parameters were:

$\dot{V}_c$ = condenser volumetric airflow rate (scfm)

T_{ci} = condenser inlet air temperature (°F)

$\dot{V}_e$ = evaporator volumetric airflow rate (scfm)

T_{ei} = evaporator inlet air temperature (°F)

T_{eo} = evaporator outlet air temperature (°F)

P_c = condenser inlet refrigerant pressure (psig)

P_e = evaporator outlet refrigerant pressure (psig).

N = compressor speed (rpm)

τ = compressor torque (in•lb)

The compressor speed and torque were measured to calculate the compressor power ($\dot{W}$).

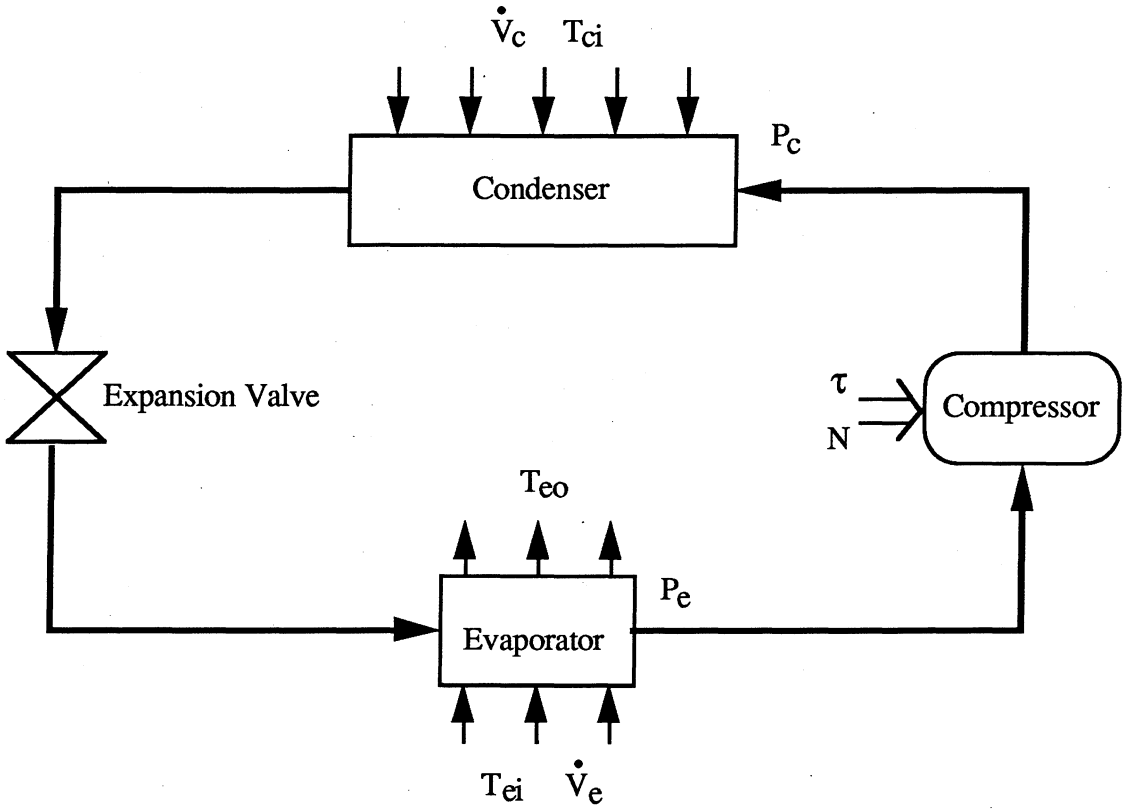

Figure 3.1. Test Facility.

Model Development Procedure

The modeling procedure is based on that presented by Crawford and Woods (1985). This procedure develops a dynamic system model from the actual system data. The general linear model form is expressed as follows:

$$y(k) = \mathbf{b}_n' \mathbf{x}_n(k) + e_n(k) \tag{3.1}$$

where y(k) is the predicted scalar variable (dependent), $\mathbf{x}_n(k)$ is the vector of n independent variables, $\mathbf{b}_n$ is the vector of n model coefficients, and $e_n(k)$ is the error scalar containing the model error at each time step (k). The independent vector $\mathbf{x}_n(k)$ may contain previous dependent values y(k-i) as well as any combination of previous and current independent variables.

The coefficient vector ($\mathbf{b}_n$) is determined by a least-squares regression procedure using the actual values of the dependent and independent variables. The error is minimized over a time interval (assumed 1) from k_i to k_f. The resulting equation for $\mathbf{b}_n$ is:

$$\mathbf{b}_n = \mathbf{W}_n^{-1} \mathbf{z}_n \tag{3.2}$$

where $\mathbf{W}_n$ (nxn) and $\mathbf{z}_n$ (nx1) are calculated from the following equations:

$$\mathbf{W}_n = \sum_{k=1}^{k_f} \mathbf{x}_n(k)\mathbf{x}_n'(k) \tag{3.3}$$

$$\mathbf{z}_n = \sum_{k=1}^{k_f} y(k)\mathbf{x}_n(k) \tag{3.4}$$

The inversion of $\mathbf{W}_n$ will become cumbersome when n becomes large. To avoid the inversion of the nxn matrix $\mathbf{W}_n$, Equations (3.2) through (3.4) are rewritten for the n+1 case. This results in:

$$\mathbf{b}_{n+1} = \mathbf{W}_{n+1}^{-1}\mathbf{z}_{n+1} \tag{3.5}$$

$$\mathbf{W}_{n+1} = \sum_{k=1}^{k_f} \mathbf{x}_{n+1}(k)\mathbf{x}_{n+1}'(k) \tag{3.6}$$

$$\mathbf{z}_{n+1} = \sum_{k=1}^{k_f} y(k)\mathbf{x}_{n+1}(k) \tag{3.7}$$

where $\mathbf{W}_{n+1}$ can be rewritten as

$$\mathbf{W}_{n+1} = \begin{bmatrix} \mathbf{W}_n & \mathbf{g} \\ \hline \mathbf{g}' & h \end{bmatrix} \tag{3.8}$$

with

$$\mathbf{g} = \sum_{k=1}^{k_f} \mathbf{x}_n(k)\mathbf{x}_{n+1}(k) \tag{3.9}$$

$$h = \sum_{k=1}^{k_f} \mathbf{x}_{n+1}^2(k) \tag{3.10}$$

The inverse of $\mathbf{W}_{n+1}$ is then found to be

$$\mathbf{W}_{n+1}^{-1} = \begin{bmatrix} \mathbf{A} & \mid & \mathbf{d} \\ \hline \mathbf{d}' & \mid & c \end{bmatrix} \tag{3.11}$$

where

$$c = (h - \mathbf{g}'\mathbf{W}_n^{-1}\mathbf{g})^{-1} \tag{3.12}$$

$$\mathbf{d} = -c\mathbf{W}_n^{-1}\mathbf{g} \tag{3.13}$$

$$\mathbf{A} = \mathbf{d}\mathbf{d}'c^{-1} + \mathbf{W}_n^{-1}. \tag{3.14}$$

The use of Equations (3.11) through (3.14) allows the calculation of $\mathbf{W}_n^{-1}$ without explicitly carrying out a matrix inversion which can become very time consuming when n becomes large. The only explicit inversion of $\mathbf{W}_n$ occurs when n = 1 and is a trivial calculation.

The significance of each of the added variables is then determined using the F-statistic:

$$F = \frac{(SSE_n - SSE_{n+1})(k_f - n - 1)}{SSE_{n+1}} \tag{3.15}$$

where SSE_n and SSE_{n+1} are the sums of the squared error calculated as follows:

$$SSE_n = \sum_{k=1}^{k_f} [y(k) - y_n'(k)]^2 \tag{3.16}$$

$$SSE_{n+1} = \sum_{k=1}^{k_f} [y(k) - y_{n+1}'(k)]^2 \tag{3.17}$$

where $y_n'(k)$ is the value predicted by the model at time k. The values of $y_n'(k)$ are calculated by:

$$y_n'(k) = \sum_{i=1}^{n} \mathbf{b}_i \mathbf{x}_i(k) \tag{3.18}$$

The F-statistic is used in the "System Model Developer" program to determine the most significant variables which may then be added to the model. Another gauge of a variable's significance is presented by Woods and Crawford (1985), the t -statistic:

$$t = \frac{\mathbf{b}_{ni}(k_f - n)}{SSE_n \mathbf{W}_{n-1}^{-1}(i,i)} \tag{3.19}$$

where t is the significance of the i^{th} independent variable. This statistic may be used for any variable at any point in the modeling process, whereas the F-statistic is only meaningful for the variable being added.

System Model Developer

A program has been written to implement the procedure described in the previous section. The "System Model Developer" (Appendix A) is a program written in TrueBasic™ that determines the coefficients of the model from the system data. This fully integrated program was used to model the overall automotive air conditioning system. Over the course of this study, features have continually been added to aid in the model development. The program is user friendly in that word commands are utilized, and the program has been made robust so that incorrect entries will not cause the program to crash (which can be time consuming because the system data would have to be reread from the disk file). The program was written in a modular form to make modifications less difficult.

The program has been configured to read comma-delimited text, which is the form used by the data reduction software used for this study. The available independent and dependent variables may be changed as necessary by manipulating the program as described in the program comments. The data is called from a file specified by the user and read into an array. The user is then prompted to enter a sampling interval. For this study, the data were collected at one-second intervals, but it was found that a five-second interval was adequate for modeling purposes. A five-second interval was entered for the model, and the program then selected every fifth data point. At this point, a menu of dependent and independent variables is shown. The user is then prompted to select a dependent variable to be modeled.

The independent variables may be selected in one of four ways: 1) The independent (or previous time step dependent) variable and its lag may be specified manually. The program will then calculate the new model coefficient(s). 2) The most significant independent variable can be determined automatically by selecting 'Auto' instead

of a variable. The program will then search through the independent variables in the menu and the previous steps of the selected dependent variable for the variable and lag that is the most significant. A table containing the most significant lag term of each of the variables and its significance is printed. The significance is determined from the F-statistic if n is greater than one, otherwise the standard deviation of error is used. 3) An independent (or the dependent) variable may be chosen and then 'Auto' selected instead of a lag. The program will find the most significant lag term, using the same criteria as in 2). 4) The user may select the 'Model Check' option from the menu. This allows a previously calculated model to be entered and run on the current data set to verify the model's applicability to new system data.

The model, t-statistics, standard deviation of error, and F-statistic are then printed on the screen. The user may then see the current model plotted with the previous model and the actual dependent variable data. This may be done in two ways. Typing 'y' will scale the graph from zero to some automatically chosen maximum. Typing 'm' will automatically choose a minimum and maximum for the y-axis based on the data and the current model values. This allows a closer view of the model. The program then asks whether or not to keep the last variable added. The independent variable process is then repeated unless 'None' is selected instead of an independent variable.

If 'None' is selected the model may then be stored in a file. The model coefficients and/or calculated values can be saved. The model coefficients are saved as a text file. The calculated values are saved as comma-delimited text along with the dependent and independent variable data at the specified time interval. A new dependent variable or data set may then be run without exiting the program.

Data Reduction Program

Before the data was loaded into the "System Model Developer" program, it was run through the "Data Reduction" program (Appendix B) to calculate the values which were not measured directly. The "Data Reduction" program also calculated many other system parameters which were useful in determining the proper operation of the air conditioning system and data acquisition equipment.

The values which were calculated and used in the modeling process were the condenser and evaporator airflow rates and the compressor power. The airflows were determined from the pressure drops across the heat exchangers. The actual airflow rates were calculated from curve fits of the pressure drop and actual flow data collected with a hot wire anemometer (Knobloch 1992). The compressor power was calculated from the speed and torque data.

The program also calculated the enthalpies of the air and refrigerant at the inlet and outlet of each of the components. The air-side enthalpies were calculated using curve fits of the data (Siambekos 1991). The condensation rate of the evaporator was estimated using the inlet and outlet relative humidities and the air property routines. The refrigerant properties were also calculated from curve fits of the data (ASHRAE 1989). From these the heat transfer of each of the components could be calculated.

The "Data Reduction" program was set up to read comma-delimited text. The data may be averaged or analyzed independently. The results can be stored in either a table form or a file of comma-delimited text. For this study, the data were analyzed independently and stored as comma-delimited text to be read into the "System Model Developer".

Chapter 4

SYSTEM MODEL

Test Plan

The parameters to be modeled by this study are the evaporator outlet air temperature (T_{eo}), the evaporator outlet pressure (P_e), the condenser inlet pressure (P_c), and the compressor power ($\dot{W}$). These dependent parameters are functions of five (independent) inputs. The independent inputs investigated here are the compressor speed (N), the evaporator inlet air temperature (T_{ei}), and the condenser volumetric airflow rate ($\dot{V}_c$). The other two parameters are the condenser inlet air temperature (T_{ci}) and evaporator volumetric airflow rate ($\dot{V}_e$). The effect of the condenser air temperature was not evaluated because the test facility (Knobloch 1992) used ambient air; no provisions for controlling the air temperature were available at the time of the test. The evaporator airflow rate was not investigated because the heating elements could provide only a fixed amount of heat. This caused the temperature to fluctuate as the airflow rate was varied under heated conditions. The airflow tests therefore had to be run using ambient air. The ambient air was not warm enough during the tests to prevent evaporator frosting.

The test plan was devised to isolate the effects of each of the independent variables. Each independent variable was varied while holding the others as constant as possible. This made the dynamics and nonlinearities of the dependent parameter clear so that a model form could be easily selected. The nonlinearities were exposed by stepping each of the dependent parameters through at least three levels, providing transients in each direction, i. e. high, low, and medium. Since nonlinearities also occured with compressor speed, the

tests were run with three compressor speeds. Each step was run long enough for the system to reach steady-state conditions.

The first dependent parameter investigated was the compressor speed. This test was run by stepping the compressor from 800 rpm to 2400 rpm and back down again in 400 rpm steps while maintaining the other parameters constant (Figure 4.1).

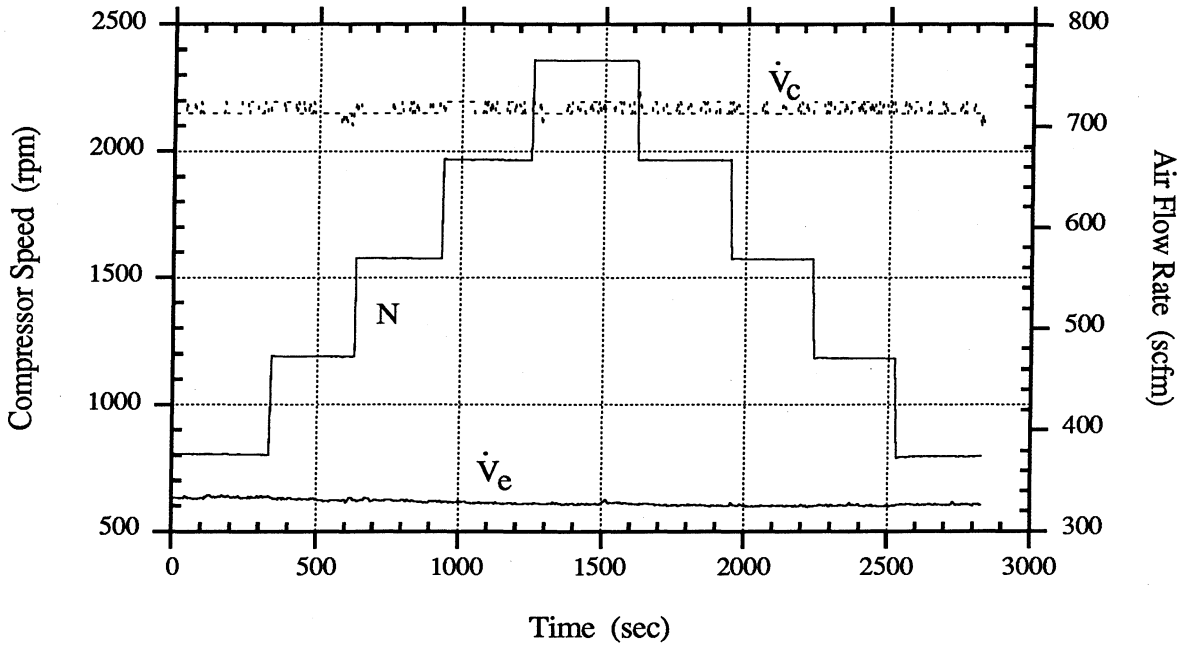

Figure 4.1. Compressor Speed and Evaporator and Condenser Airflow Rates.

Figure 4.2 shows that the inlet air temperatures rose toward the end of the test. The evaporator inlet air temperature rose about 5°F in the last 25 minutes of the test, while the condenser inlet air temperature increased about 2°F. The condenser temperature did not affect the results much, causing a slight increase in the condenser pressure. The evaporator temperature rise affected the outlet air temperature very noticeably, and its effect is shown in the model.

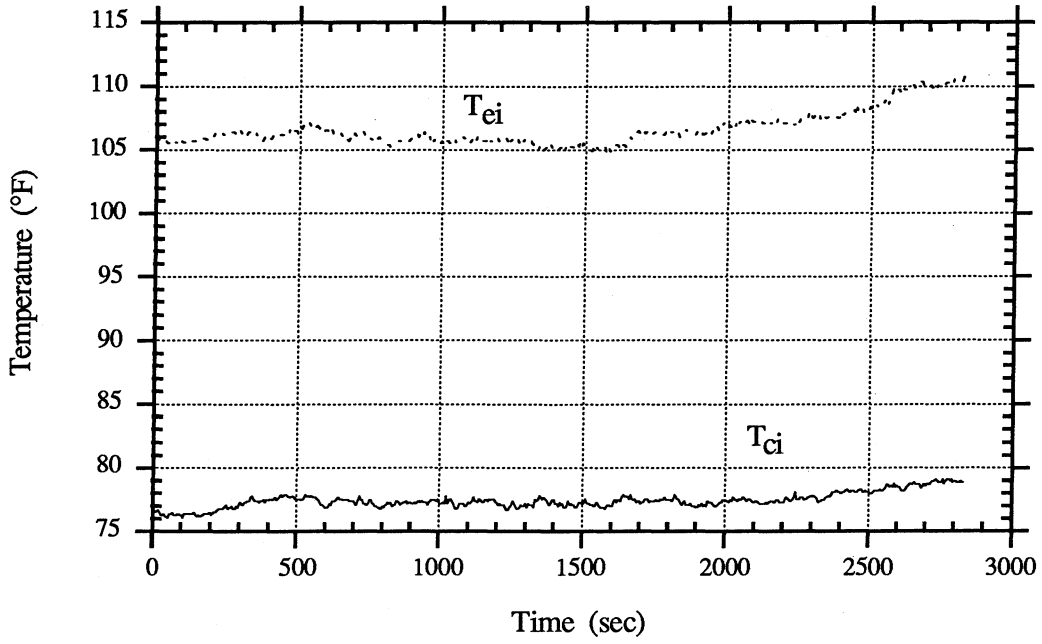

Figure 4.2. Condenser and Evaporator Inlet Air Temperatures.

The effect of the evaporator inlet air temperature was evaluated next. The temperature was stepped over a range of about 12°F (Figure 4.3). The compressor speed was stepped in 800 rpm decrements from 2400 to 800 rpm (Figure 4.4), repeating the temperature steps for each speed.

The last test investigated the effect of the condenser airflow rate. This was similar to the inlet air temperature test, except the condenser airflow rate (Figure 4.5) was changed between 1150 and 550 scfm, with an intermediate step at 850 scfm. The compressor speed was stepped from 2400 to 800 rpm as before. The inlet air temperatures were kept constant (Figure 4.6).

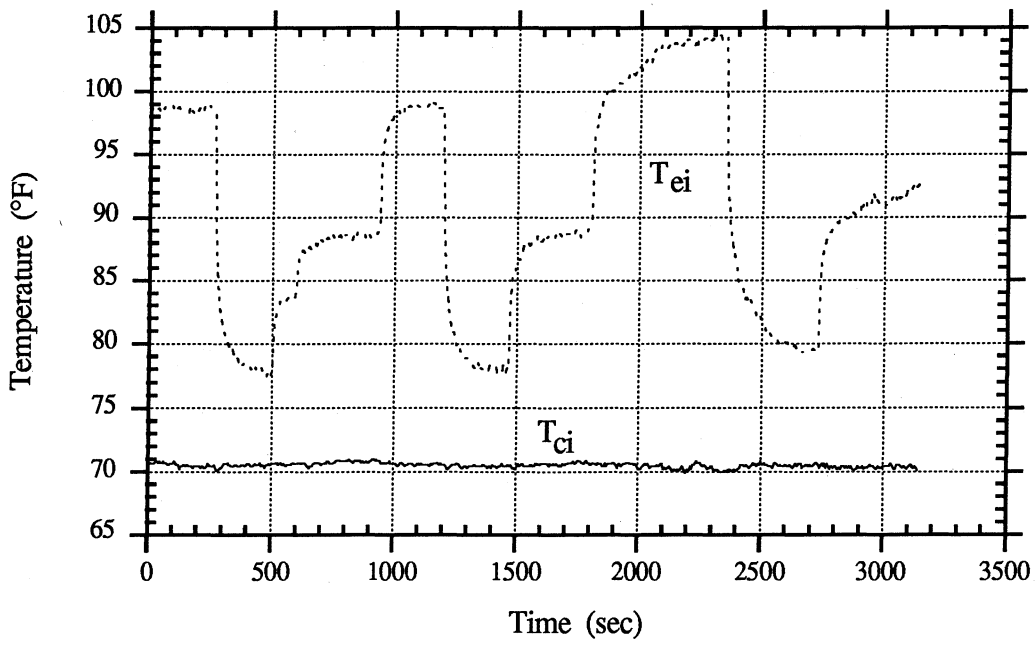

Figure 4.3. Condenser and Evaporator Inlet Air Temperatures.

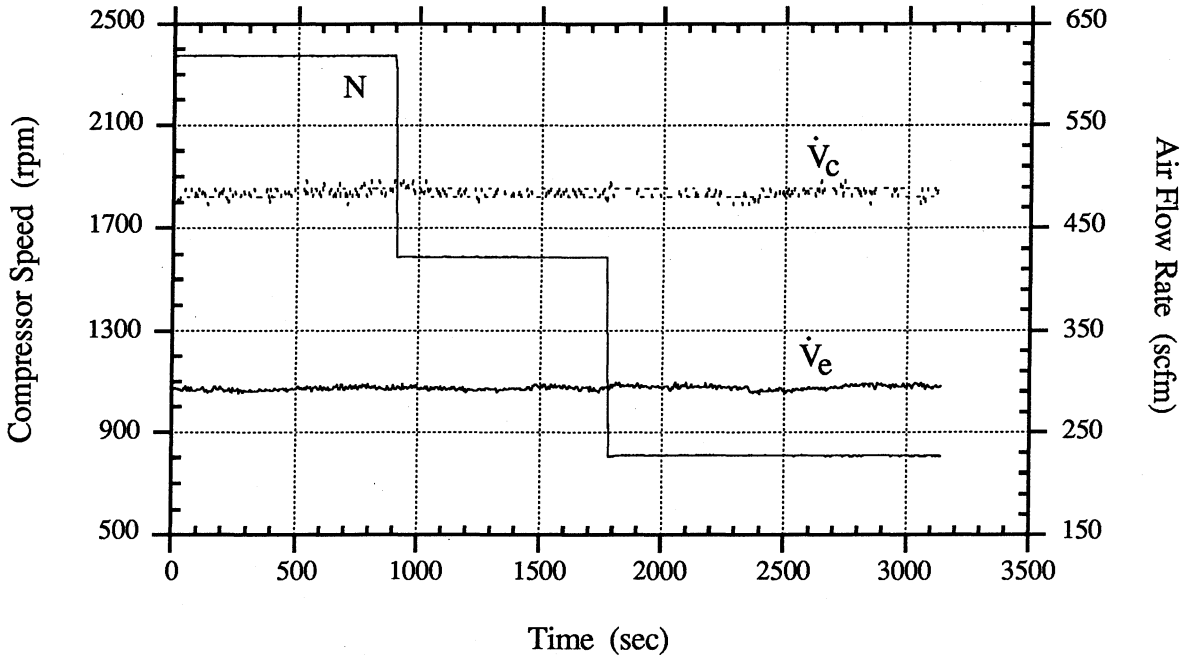

Figure 4.4. Compressor Speed and Condenser and Evaporator Airflow Rates.

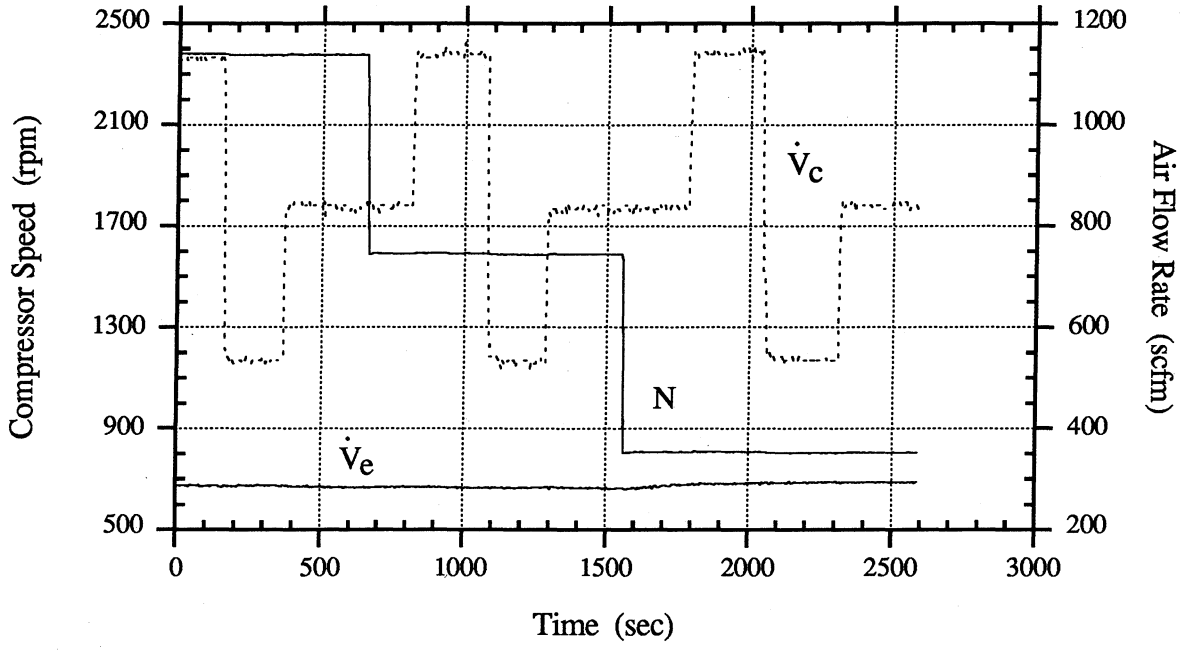

Figure 4.5. Compressor Speed and Condenser and Evaporator Airflow Rates.

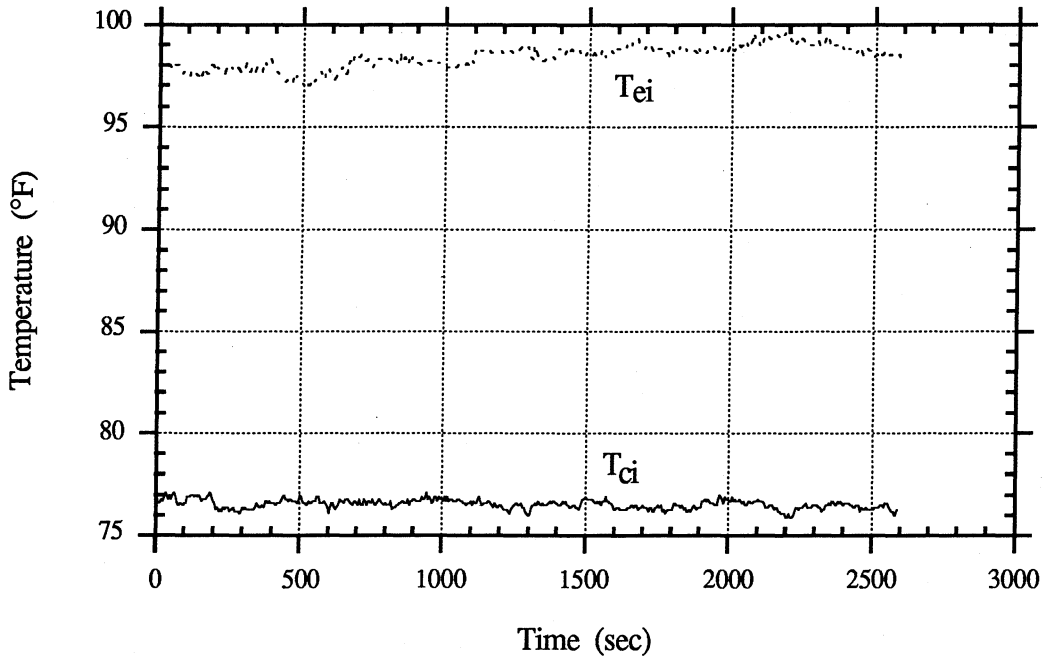

Figure 4.6. Condenser and Evaporator Inlet Air Temperatures.

Compressor Speed

The "System Model Developer" program was used to determine the significant variables in the models. Since the compressor speed was the only parameter varied in this test, it was expected to be the only independent term in the models. All the dependent terms were found to be nonlinear, second-order functions of the compressor speed with a constant term. A sampling interval of 5 seconds provided adequate resolution for all of the dependent variables.

The condenser pressure model was determined to be:

$$P_c(k) = (0.815)P_c(k\text{-}1) + (0.0158 \text{ psig/rpm})N(k) - (2.40\text{x}10^{-6} \text{ psig/rpm}^2)N^2(k) + 15.9 \text{ psig} \quad (4.1)$$

where $P_c(k\text{-}1)$ is the condenser pressure at the previous time step, and N(k) is the compressor speed at the current time. Figure 4.7 shows the model and the data. The pressure term at the previous time step provides the smooth first order transient response. The time constant was calculated by transforming Equation 4.1 into the z-domain. This results in the following:

$$P_c(z) = \frac{z}{z - r} U(z) \quad (4.2)$$

where U(z) is a constant containing the independent terms. The magnitude of the root (r = 0.815) of the denominator was then used to find the time constant (τ_c) with the equation:

$$r = e^{-T/\tau_c} \quad (4.3)$$

where T is the sampling interval (5 seconds). The time constant was found to be 24 seconds. The time constant is a useful measure of a system's transient response. The model had a standard deviation of 1.4 psig. This error could be reduced if the increase in condenser air temperature in the last 25 minutes of data was accounted for. The increase in temperature (Figure 4.2) raised the condenser pressure slightly for that part of the test and contributed to the steady-state errors.

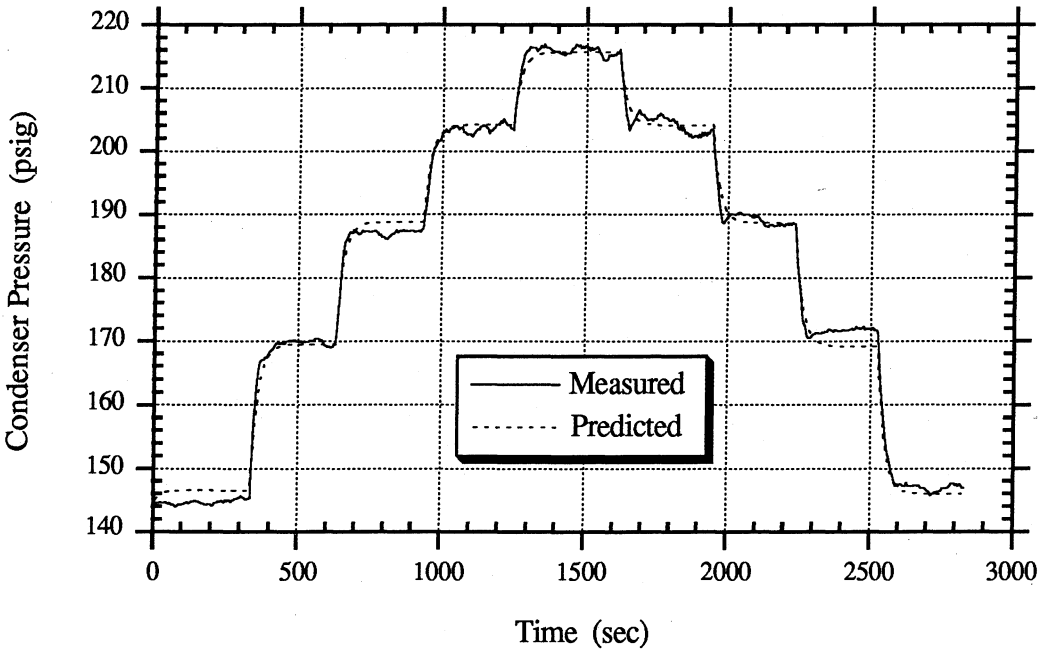

Figure 4.7. Condenser Pressure.

The evaporator outlet pressure changes were more abrupt, and did not require the previous pressure term (the time constant was almost 0) in the model. The evaporator pressure was found to be an inverse function of the compressor speed. The model is:

$$P_e(k) = (1.17 \times 10^4 \text{ psig•rpm})/N(k) - (4.07 \times 10^6 \text{ psig•rpm}^2)/N^2(k) + 18.6 \text{ psig.} \quad (4.4)$$

This model has a standard deviation of 0.23 psig (Figure 4.8). The data appear to have some quantization error. This is due to the data acquisition software rounding the data to the nearest tenth, not due to any measurement limitations. The evaporator pressure is noisy because of the small refrigerant mass in the evaporator and its close proximity to the compressor.

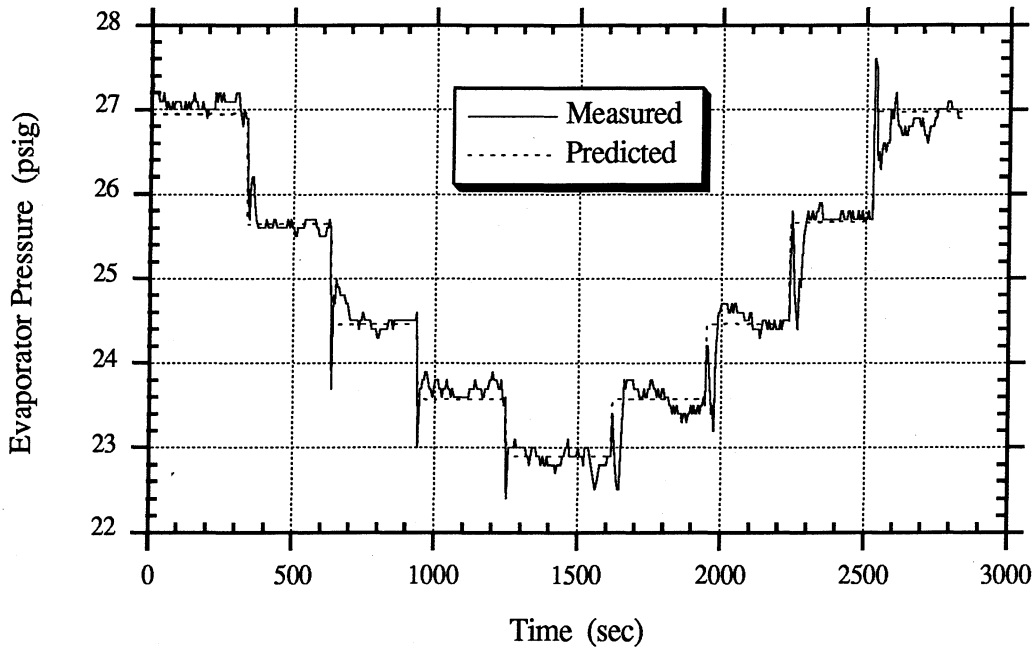

Figure 4.8. Evaporator Pressure.

The evaporator outlet temperature has a second-order transient response (the temperature overshoots the steady state values) which is reflected in the model by the two previous temperature terms. The model is:

$$T_{eo}(k) = (1.33)T_{eo}(k\text{-}1) - (0.516)T_{eo}(k\text{-}2) + (5340\ ^\circ F \cdot rpm)/N(k)$$
$$- (1.25 \times 10^6\ ^\circ F \cdot rpm^2)/N^2(k) + (0.148)T_{ei}(k) - 8.65\ ^\circ F \qquad (4.5)$$

where T_{ei} is the evaporator inlet air temperature. This term appears in the model because it increased about 5°F over the course of the test and had a large impact on the outlet air temperature, as can be seen in Figure 4.9. The standard deviation of this model is 0.48°F.

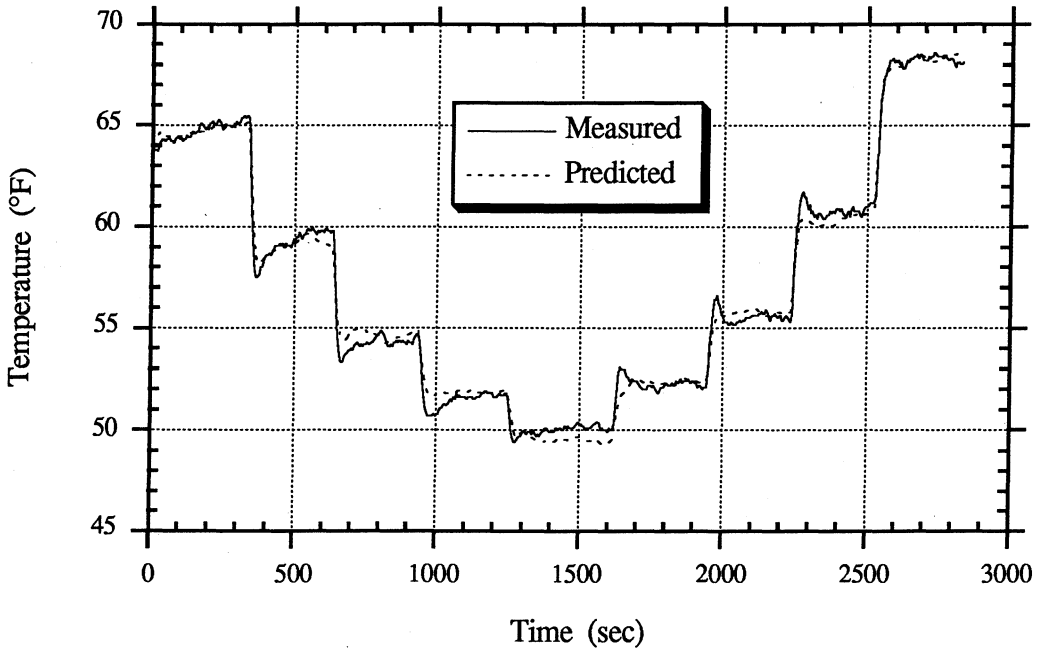

Figure 4.9. Evaporator Outlet Air Temperature.

The time constants for this second-order system were calculated by first transforming the equation into the z-domain.

$$T_{eo}(z) = \frac{z^2}{z^2 - 1.33z + 0.516} U(z) \qquad (4.6)$$

The characteristic equation (the denominator) has complex roots. The resulting function is an exponentially damped sinusoid. The general form of solution for this equation (Franklin 1980) with a step input is:

$$T_{eo}(t) = A + Be^{-t/\tau_d}\cos(2\pi T/\tau_p + \phi) \tag{4.7}$$

where τ_d is the time constant and τ_p is the oscillation period. τ_d and τ_p are calculated from the characteristic equation by setting:

$$e^{-2T/\tau_d} = 0.516 \tag{4.8}$$

$$2e^{-T/\tau_d}\cos(2\pi T/\tau_p) = 1.33. \tag{4.9}$$

Solving these equations gives a time constant of 15 seconds and an oscillation period of 81 seconds.

Inlet Air Temperature

The condenser pressure was tested and found to be a very weak function of the evaporator inlet air temperature. Figure 4.10 reveals that the condenser pressure was very steady for this test, not reflecting the steps in the evaporator air temperature. The "System Model Developer" program confirmed this conclusion, giving the most significant temperature term a F-statistic of only 15 (anything less than 100 improves the model only marginally). The evaporator inlet temperature was therefore not included in the condenser model.

The effect of the evaporator inlet air temperature on the evaporator pressure was dependent on the compressor speed. The T_{ei}/N term in the model reflects this relationship. Since the compressor speed was changed in this test, the compressor terms determined previously were included in the model.

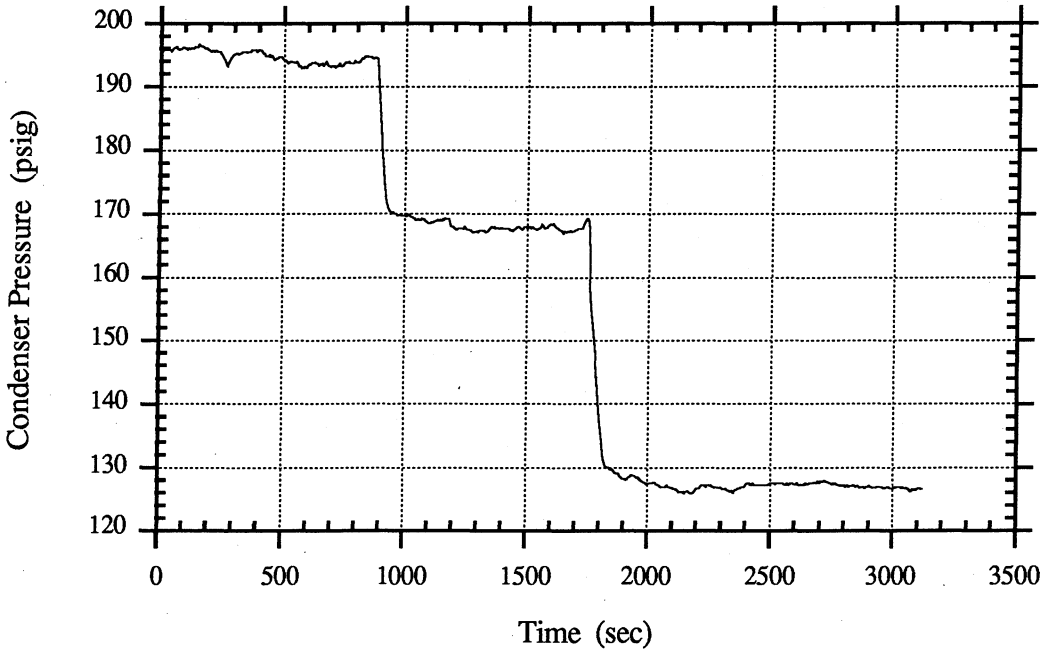

Figure 4.10. Condenser Pressure.

$$P_e(k) = (1.43\text{x}10^4 \text{ psig•rpm})/N(k) - (3.96\text{x}10^6 \text{ psig•rpm}^2)/N^2(k) + (6.47\text{x}10^{-2} \text{ psig/°F})T_{ei}(k) - (46.6 \text{ psig•rpm/°F})T_{ei}/N(k) + 7.20 \text{ psig} \quad (4.10)$$

where T_{ei} is the evaporator inlet air temperature. This model (Figure 4.11) followed the pressure dynamics well, and had a standard deviation of 0.17 psig.

The evaporator outlet air temperature was a strong function of the inlet air temperature, particularly at low compressor speeds. The model included the T_{ei}/N term to satisfy the dependence on compressor speed as well as a straight inlet air term.

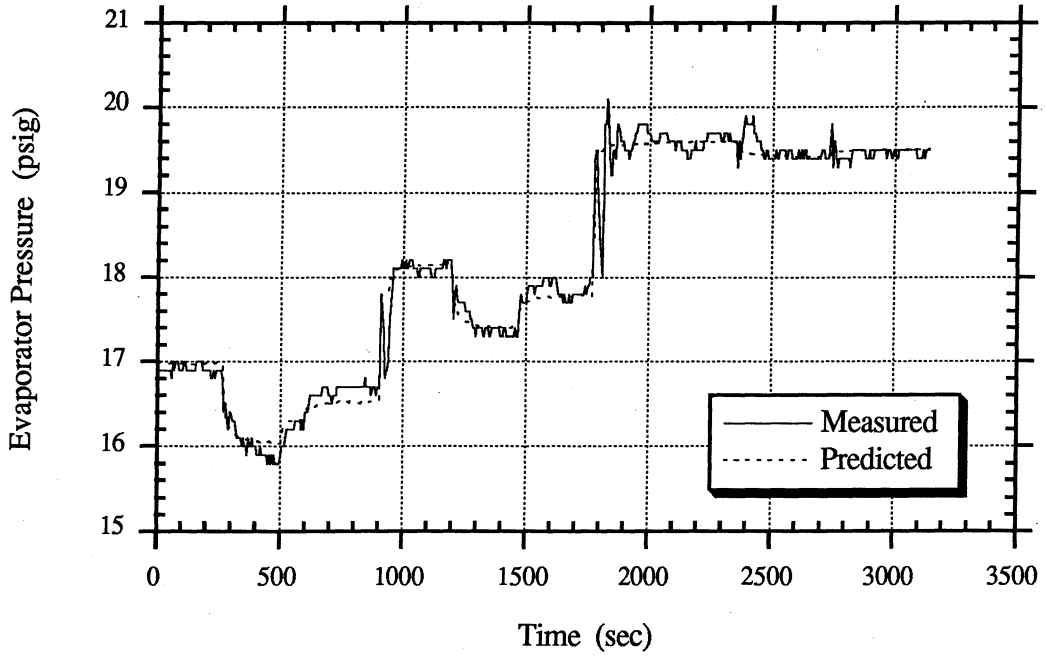

Figure 4.11. Evaporator Pressure.

$$T_{eo}(k) = (1.08)T_{eo}(k\text{-}1) - (0.329)T_{eo}(k\text{-}2) - (9.03\text{x}10^{3}\ ^{\circ}\text{F}\bullet\text{rpm})/N(k) + (3.14\text{x}10^{6}\ ^{\circ}\text{F}\bullet\text{rpm}^{2})/N^{2}(k) + (93.6\ \text{rpm}^{2})T_{ei}/N(k) + (6.84\text{x}10^{-2})T_{ei}(k) + 2.37\ ^{\circ}\text{F} \quad (4.11)$$

All the inlet temperature terms are at the current time step because there is very little thermal capacitance in the evaporator. The model tracked the data (Figure 4.12) with a standard deviation of 0.31°F. The time constant and oscillation period are 9 and 91 seconds, respectively.

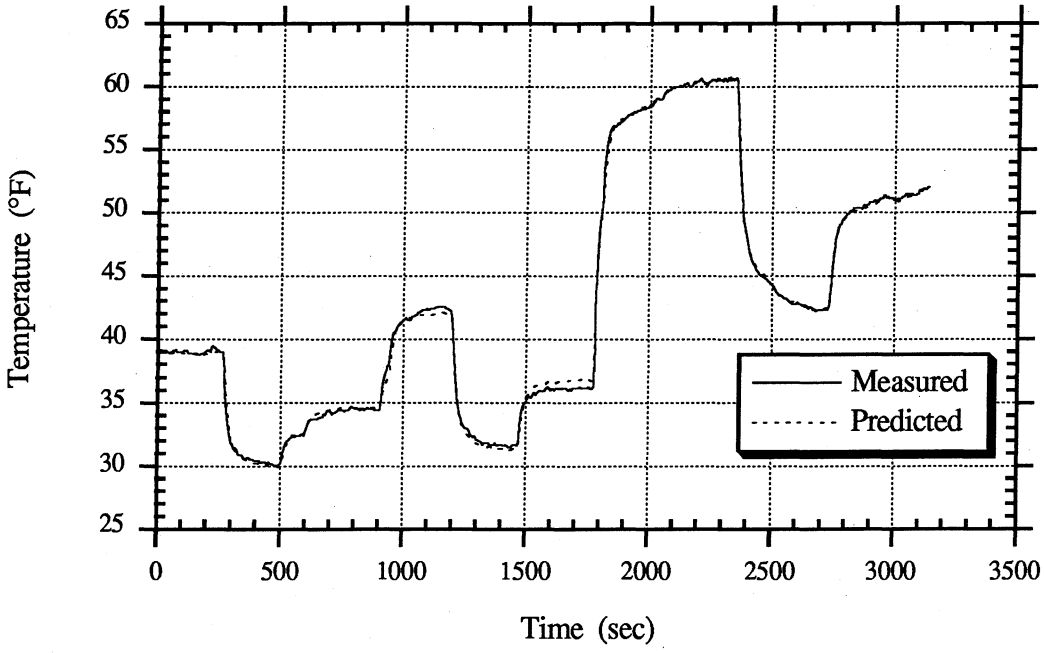

Figure 4.12. Evaporator Outlet Air Temperature.

The compressor power (Figure 4.13) was a weak function of the inlet air temperature. The compressor power could be modeled with a standard deviation of 18.9 Btu/hr by using the compressor speed terms:

$$\dot{W}(k) = [2.66\ (\text{Btu/hr})/\text{rpm}]N(k) - [1.48\text{x}10^{-4}\ (\text{Btu/hr})/\text{rpm}^2]N^2(k) - 494\ \text{Btu/hr} \qquad (4.12)$$

The two most significant inlet air temperature terms improved the standard deviation by 4 Btu/hr to 15.9 Btu/hr, which was not considered to be a worthwhile improvement in this case.

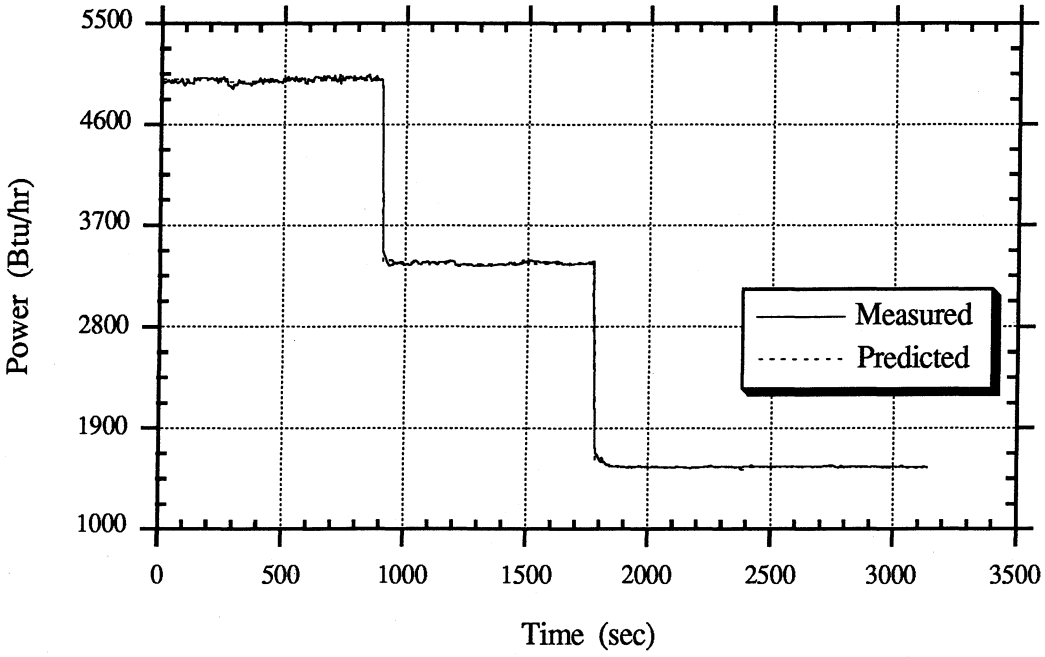

Figure 4.13. Compressor Power.

Condenser Air Flow

Condenser pressure was found to be an inverse second-order function of the airflow rate. The effect of the airflow rate increased as the compressor speed and heat transfer requirements increased, so the $N/\dot{V}_c$ terms were introduced into the model.

$$P_c(k) = (0.776)P_c(k-1) + (9.69\times10^{-3}\ \text{psig/rpm})N(k) - (2.40\times10^{-6}\ \text{psig/rpm}^2)N^2(k) + (4.45\ \text{psig}\cdot\text{scfm/rpm})N/\dot{V}_c(k) - (121\ \text{psig}\cdot\text{scfm}^2/\text{rpm})N/\dot{V}_c^2(k) + 18.2\ \text{psig} \quad (4.13)$$

The condenser pressure (Figure 4.14) could be predicted by this model within 1.31 psig. The time constant was 20 seconds.

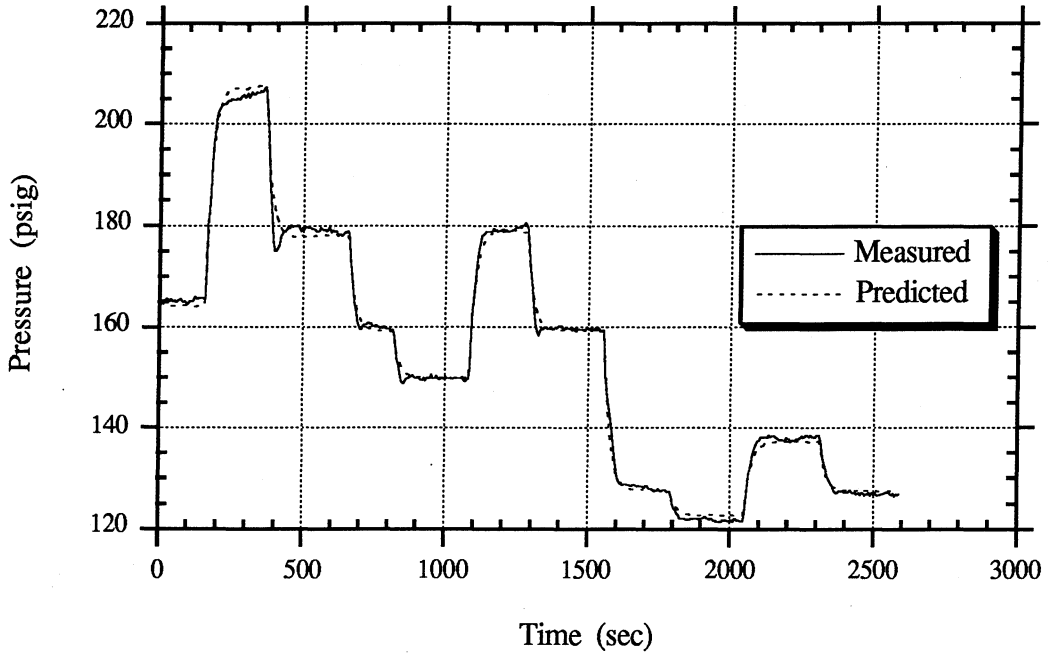

Figure 4.14. Condenser Pressure.

The effect of condenser airflow rate on the evaporator pressure was similar to the condenser pressure, though much less significant. The evaporator pressure (Figure 4.15) was affected much more at high compressor speeds because of the large condenser pressure changes caused by the condenser airflows. The evaporator pressure is noisier than the condenser pressure due to the smaller refrigerant mass (it is also a vapor) and the proximity of the evaporator to the compressor. The model had a standard deviation of 0.35 psig.

$$P_e(k) = (1.88\text{x}10^4 \text{ psig•rpm})/N(k) - (7.69\text{x}10^6 \text{ psig•rpm}^2)/N^2(k) + (0.656 \text{ psig•scfm/rpm})N/\dot{V}_c(k\text{-}1) - (485 \text{ psig•scfm})/\dot{V}_c(k\text{-}1) + 10.0 \text{ psig} \quad (4.14)$$

The condenser airflow terms are lagged by one time step to account for the refrigerant transportation delay between the condenser and evaporator.

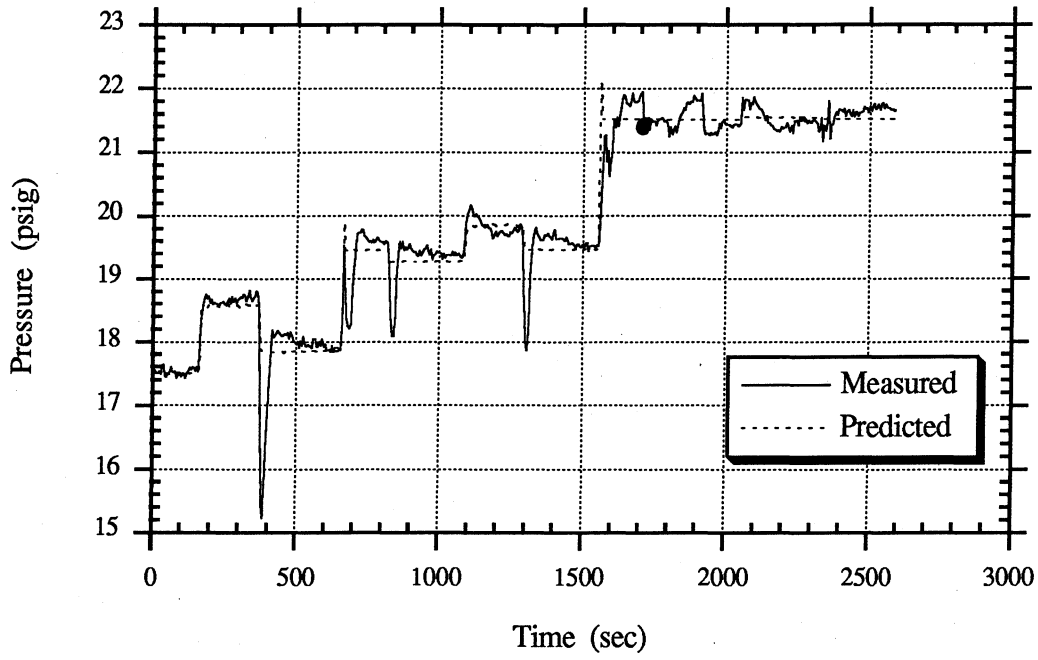

Figure 4.15. Evaporator Pressure.

The three peaks after 1500 seconds (the 800 rpm region) seem to correspond initially to the condenser airflow changes, then drop back to the previous value. The peaks last about 1 minute, which at this compressor speed corresponds to the refrigerant circulation time.

The compressor power (Figure 4.16) was also a second-order function of the condenser airflow rate, with the effect diminishing with the compressor speed. The power changed in response to the changes in pressure rise required when the condenser pressure changed. Only the $N/\dot{V}_c^2$ term was found to be significant in this model.

$$\dot{W}(k) = [4.32\ (\text{Btu/hr})/\text{rpm}]N(k) - [2.40\text{x}10^{-4}\ (\text{Btu/hr})/\text{rpm}^2]N^2(k) + [8.59\text{x}10^4\ (\text{Btu/hr})\cdot\text{scfm}^2/\text{rpm}]N/\dot{V}_c^2(k) - 806\ \text{Btu/hr} \quad (4.15)$$

This model had a standard deviation of 49 Btu/hr.

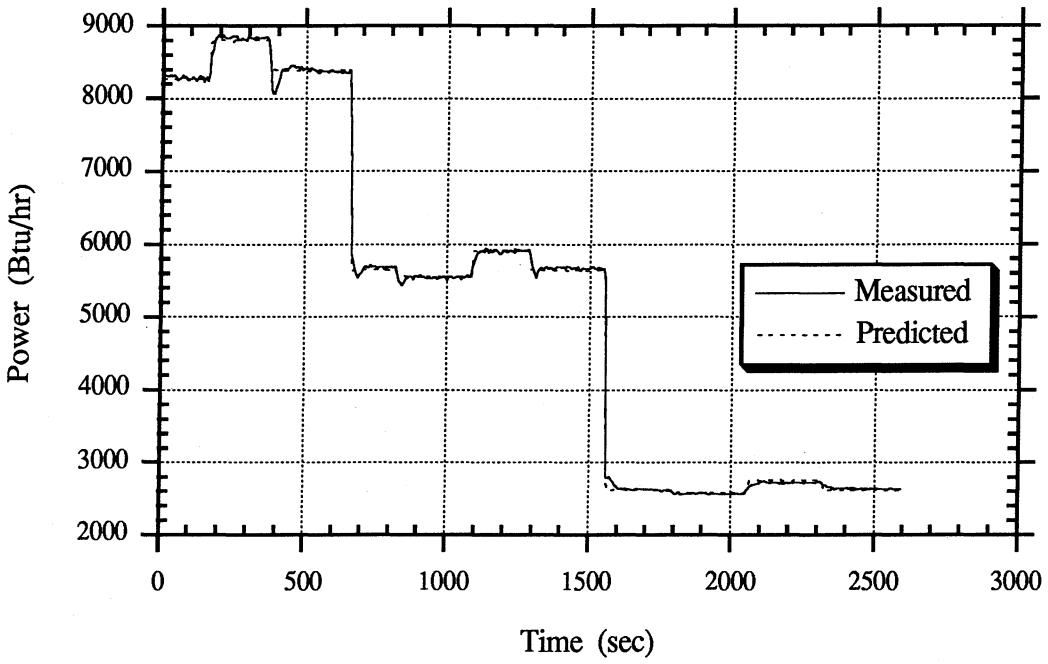

Figure 4.16. Compressor Power.

The evaporator outlet air temperature (Figure 4.17) was found to have a simple $1/\dot{V}_c$ relationship delayed by 6 time steps. This lag was a result of the refrigerant transportation delay. The 6 time steps correspond to a 30 second delay. The average refrigerant flow rate during this test was 333 lb/hr with a total charge of 2.6 lb of refrigerant and oil. Dividing the mass flow rate by the total charge gives an average refrigerant circulation time of 28 seconds. This indicates that the refrigerant had to circulate a little more than one time to change the air temperature. The model is:

$$T_{eo}(k) = (1.10)T_{eo}(k-1) - (0.234)T_{eo}(k-2) + (881\ ^\circ F \bullet rpm)/N(k) - (8.95 \times 10^5\ ^\circ F \bullet rpm^2)/N^2(k) + (278\ ^\circ F \bullet scfm)/\dot{V}_c(k-6) + (3.14 \times 10^{-2}\ ^\circ F/scfm)\dot{V}_e(k) - 4.84\ ^\circ F \qquad (4.16)$$

The evaporator airflow rate term ($\dot{V}_e$) is included in this model because the evaporator airflow (Figure 4.5) changed significantly during the last part of the test, creating a steady-state error. The inclusion of this term in the model corrected the error. The resulting standard deviation of this model is 0.44°F. The roots were 0.81 and 0.29 which resulted in time constants of 6 and 17 seconds, respectively. There is no oscillation in this case, just two exponentially-decaying terms.

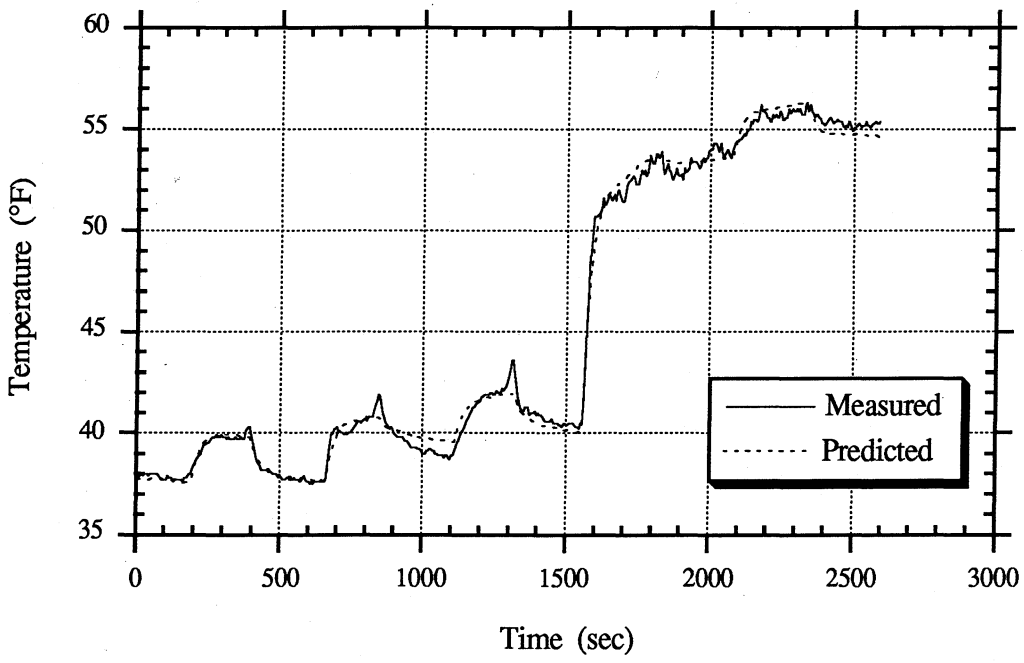

Figure 4.17. Evaporator Outlet Air Temperature.

Chapter 5

CONCLUSION

Summary

The objective of the current study was to develop a low-order empirical system model of an automotive air conditioning system. The model had to track the system transients, since automotive air conditioning systems rarely operate at steady state. The modeling technique chosen for this was based on an auto-regressive, least-squares approach, and the program to implement it was presented.

The test plan to isolate each of the parameters was then discussed. The tests each involved at least three steps in increasing and decreasing directions. This was to expose any nonlinearities that may have been present. The compressor speed was also changed during each test since the effects of some of the independent variables on the dependent variables were also functions of the compressor speed. The independent variables not being tested were held as constant as possible, though on a couple of occasions they wandered from a steady value. These effects were accounted for in the models, where necessary.

The effect of the compressor speed on each of the dependent parameters was found to be a second-order relationship. The condenser pressure model included one transient term and had a standard deviation of error of 1.4 psig. The evaporator pressure and temperature were found to be inversely related to the compressor speed, so the speed terms were included as 1/N in the model. The evaporator model did not include any transient

terms due to its fast response. The evaporator model had a standard deviation of 0.23 psig. The evaporator outlet air temperature model included two transient terms to model its underdamped response. The outlet air temperature model had a standard deviation of 0.48°F.

The evaporator inlet air temperature did not affect all of the dependent parameters significantly. The condenser pressure and compressor power were not found to be strong functions of the inlet air temperature. The evaporator pressure was significantly affected, and two terms were added to model this effect. The effect varied with compressor speed, so a T_{ei}/N term was used along with the T_{ei} term. This model resulted in a standard deviation of 0.17 psig. The outlet air temperature was modeled using the same terms. This model followed the temperature dynamics very well and had a standard deviation of 0.31°F.

The next independent variable investigated was the condenser airflow rate. This variable affected all of the dependent variables. The condenser pressure was affected most, being a second-order function of $N/\dot{V}_c$. This relationship arose out of the pressure's changing response to the airflow rate with compressor speed. The addition of these terms modeled the condenser pressure with a standard deviation of 1.31 psig. The evaporator pressure was moderately affected by the condenser airflow rate. The relationship was not second order, however, and only the $N/\dot{V}_c$ and $1/\dot{V}_c$ terms were required (with a lag of one time step to account for refrigerant transportation). The model had a standard deviation of 0.35 psig. The evaporator outlet air temperature had even more transport delay in response to the condenser airflow, requiring 6 time steps. The temperature response to airflow rate was not a strong function of the compressor speed and only a $1/\dot{V}_c$ term was required to achieve a standard deviation of 0.44°F. The power was also a moderate function of the

airflow rate, requiring a $N/\dot{V}_c^2$ term to model the effect. The standard deviation of error for this model was 49 Btu/hr.

Recommendations

There are several things which should be done for future studies. The test matrix was only 80% complete. The remaining independent variables were not investigated for this study because they could not be controlled. The condenser inlet air temperature needs to be controllable. It probably has a very significant effect on the system variables investigated in this study. A heater has recently been placed in the inlet duct, though the only control available at the current time is on/off, which would not be adequate to expose the response nonlinearities. A variable heater control is being studied at this time which should alleviate this difficulty. A better method of heating the evaporator inlet air temperature should also be investigated. The current arrangement does not allow the evaporator airflow rate to be changed independently from the inlet air temperature. Only a fixed amount of heat is available, so when the airflow rate is changed, the temperature also changes, obscuring the effect of the airflow changes.

Since the models in this study include the effects of only two independent variables at a time, the models are primarily of use for their form rather than the actual model coefficients. To determine the overall model coefficients, tests need to be run in which each of the independent variables is changed at least once. This overall test matrix should include all the different combinations of independent variables to determine the coefficients for accurate system models.

Other transients should also be investigated. The start-up and shutdown transients would probably be of interest in an advanced controller. The start-up transients in particular would be critical for the advanced controller to predict, since they are typically the periods of greatest system change and power consumption. Evaporator frosting effects should also be investigated. The controller would need to be able to determine the level of frosting to prevent accumulation.

REFERENCES

ASHRAE. 1989. *1989 ASHRAE Handbook--Fundamentals*, SI ed., American Society of Heating, Refrigerating, and Air Conditioning Engineers, Inc.

Cecchini, C., and Marchal, D., "A Simulation Model of Refrigerating and Air-Conditioning Equipment Based on Experimental Data", *ASHRAE Transactions*, Vol. 97, Part 2, 1991.

Crawford, R. R., and Woods, J. E., "A Method for Deriving a Dynamic System Model from Actual Building Performance Data", *ASHRAE Transactions*, Vol. 91, Part 2, 1985, pp. 1859-1874.

Davis, G. L., Chianese, F., and Scott, T. C., "Computer Simulation of Automotive Air Conditioning - Components, System, and Vehicle", 1972 SAE Congress Paper 720077.

Davis, G. L., and Scott, T. C., "Component Modeling Requirements for Refrigeration System Simulation", *Proceedings of the 1976 Compressor Technology Conference*, Purdue University, pp. 401-408.

Dhar, M., and Soedel, W., "Transient Analysis of a Vapor Compression Refrigeration System: Part 1 - The Mathematical Model", *15th International Congress of Refrigeration*, Vol. 2,Venice, Sept. 23-29, 1979.

Dhar, M., and Soedel, W., "Transient Analysis of a Vapor Compression Refrigeration System: Part 2 - Computer Simulation and Results", *15th International Congress of Refrigeration*, Vol. 2,Venice, Sept. 23-29, 1979.

Franklin, G. F., and Powell, J. D., *Digital Control of Dynamic Systems*, Addison-Wesley, Massachusetts, 1980, p. 305.

Josiassen, N. J., "Simulation of Condition Sequence During Start-up of an Evaporation Refrigerating System", *Proceedings of the 1978 Purdue Compressor Technology Conference*, July 19-21, pp. 309-316.

Knobloch, L. A., "Dynamic Modeling of an Automotive Air Conditioning Compressor from Actual Performance Data", Master's Thesis, University of Illinois at Urbana-Champaign, 1992.

MacArthur, J. W., "Analytical Representation of the Transient Energy Interactions in Vapor Compression Heat Pumps", ASHRAE Transactions, Vol. 90, Part 1B, 1984, pp. 982-996.

Michael, T. A., "Design of an Automotive Air Conditioning Test Stand for Screening and Transient Studies", Master's Thesis, University of Illinois at Urbana-Champaign, 1989.

Shirey, D. B., "Dynamic Modeling of a Residential Air-Source Heat Pump from Actual System Performance Data", Master's Thesis, University of Illinois at Urbana-Champaign, 1987.

Siambekos, C. T., "Two-Zone Modeling of a Mobile Air Conditioning Plate-Fin Evaporator", Master's Thesis, University of Illinois at Urbana-Champaign, 1991, pp. 144-149.

APPENDIX A

System Model Developer

TrueBasic™ Version 2.02

Developed on MacIntosh IIci

```
!Chrysler System Model Developer
!Automated Version 1b
!Charles Schenk
!University of Illinois
!12-8-91

! This program will read a comma-delimited text file, load the desired values into arrays
! (dependent and independent), and calculate the model coefficients. The program is
! currently configured to read the output of the "Data Reduction" Program. The parts of the
! program which need to be modified to change the modeling variables are marked with **.


! Main Program

dim dependent(1,1),independent(1,1),yd(1),id$(20),prev(20)
dim xn(1,1),xn1(1,1),wni(1,1),wi(1,1),zn(1,1),zn1(1,1),bnte(1,1)
dim bn(1,1),bn1(1,1),t(20),tt(1),bnt(1,1),inda(1)

let indmax=14                    !** This is the number of independent variables + 1. **
let nwst$="y"
do while nwst$<>"n"
  input prompt "Sound with result? ":snd$
  let snd$=lcase$(snd$[1:1])
  call indat(kft,lagt,dependent,independent,snd$,indmax)
  let lagt=-abs(lagt)
  let new$="y"
  let indep$="y"

  do while new$<>"n"
    let n=0
    print
    input prompt "Time interval? ":stp
    let kf=int(kft/stp)
    let lag=int(lagt/stp)
    call varlist
    call dselect(kf,lag,stp,d$,dependent,independent,yd)

    let newind$="y"
```

```
    call indselect(n,kf,lag,stp,id$,d$,prev,ind,inda,independent,bn,bn1,xn,xn1,yd,no,
                        newind$,auto$,indmax,man)

   do while newind$<>"n"
    if man<>1 then
      if auto$="y" then
         call auto(n,kf,lag,stp,ind,inda,no,ssen,msen,prev,independent,yd,xn,xn1,zn,
                        zn1,bn,bn1,wi,wni,id$,d$,indmax)
      end if
      let mank=0
       call x1(kf,prev(n+1),stp,ind,d$,xn1,independent)

      if n<>0 and man<>1 then
          call winv(n,kf,stp,xn,xn1,yd,wni,wi,zn,zn1,bn1,bn,no)
         call ftest(n+1,kf,stp,yd,inda,prev,independent,bn1,f,msen1,ssen,ssen1,d$)
        call ttest(n,msen1,bn1,wi,t)
      else
         call first(kf,stp,prev(n+1),ind,inda,xn1,independent,yd,zn,wni,bn,bn1,ssen1,
                        msen,msen1,d$)
        mat xn=xn1
        mat wi=wni
        call ttest(n,msen1,bn1,wi,t)
      end if
    else
      let newind$="n"
       call ftest(n,kf,stp,yd,inda,prev,independent,bn1,f,msen1,ssen,ssen1,d$)
      mat bn=bn1
      let ind=inda(n)
      let n=n-1
      let mank=1
    end if

     call result(n,kf,lag,stp,id$,prev,d$,bn,bn1,f,t,msen,msen1,ssen,ssen1,ft,tt,
                        msent,msen1t,bnt,bnte,snd$,xn,xn1,yd,independent,ind,inda,man)
    if mank<>1 then
      call varlist
       call indselect(n,kf,lag,stp,id$,d$,prev,ind,inda,independent,bn,bn1,xn,xn1,
                        yd,no,newind$,auto$,indmax,man)
    else
      let newind$="n"
    end if
  loop

   if n<>0 then call keep(n,kf,stp,lag,id$,prev,d$,bnt,bn1,f,t,msent,msen1t,yd,
                              xn,independent,inda,mank)
  print
  input prompt "Try new dependent variable? ":new$
  let new$=new$[1:1]
  let new$=lcase$(new$)
  clear
loop
input prompt "Run new data set? ":nwst$
let nwst$=nwst$[1:1]
let nwst$=lcase$(nwst$)
```

```
loop

end


Sub first(kf,stp,i,ind,inda(),xn(,),independent(,),yd(),zn(,),wni(,),bn(,),
                bn1(,),ssen1,msen,msen1,d$)

! This routine calculates the first  (n=1) model coefficients (bn and bn1) and the error
! (ssen1).


    dim bnz(1,1),wn(1,1),bnt(1,1),xnt(1,1),yc(1,1),pre(1)
   mat zn=zer(1,1)
   mat wn=zer(1,1)
   let pre(1)=i

   for k=1 to kf
      let wn(1,1)=wn(1,1)+xn(k,1)^2
      let zn(1,1)=zn(1,1)+yd(k)*xn(k,1)
   next k

   mat wni=inv(wn)
   mat bn=wni*zn
   mat bn1=bn

   let ssen1=0

   call model(yc,bn1,1,kf,stp,inda,independent,pre,d$)
   for k=1 to kf
     let ssen1=ssen1+(yc(1,k)-yd(k))^2
   next k

   let msen1=ssen1/(kf-1)
   let msen=msen1
end sub



Sub  winv(n,kf,stp,xn(,),xn1(,),yd(),wni(,),wi(,),zn(,),zn1(,),bn1(,),bn(,),no)
    dim cw(1,1),d(1,1),dt(1,1),ddt(1,1),a(1,1),g(1,1),xnxn1(1,1)
    dim gt(1,20),gwg(1,1),gw(1,20),zp(20,1),yxn(20,1),tempx(1,1)

! Winv calculates the inverse of W_{n+1} and the new model coefficients (bn1).


  let h=0
  if n>1 and no<>n then
    mat wni=wi
    mat zn=zn1
  end if
  let no=n
```

```
    mat g=zer(n,1)
    mat gt=zer(1,n)
    mat gw=zer(1,n)
    mat xnxn1=zer(n,1)
    mat cw=zer(n,n)
    mat d=zer(n,1)
    mat dt=zer(1,n)
    mat ddt=zer(n,n)
    mat a=zer(n,n)
    mat zp=zer(n,1)
    mat yxn=zer(n,1)
    mat zn1=zer(n+1,1)

    for k=1 to kf
      for i=1 to n
        let xnxn1(i,1)=xnxn1(i,1)+xn(k,i)*xn1(k,1)
        let yxn(i,1)=yxn(i,1)+yd(k)*xn(k,i)
      next i
      let h=h+xn1(k,1)^2
      let zn1(n+1,1)=zn1(n+1,1)+yd(k)*xn1(k,1)
    next k

    mat zp=yxn
    mat g=xnxn1
    mat gt=trn(g)
    mat gw=gt*wni
    mat gwg=gw*g
    let hdif=h-gwg(1,1)
    if hdif=0 then let hdif=.001
    let c=1/(hdif)
    mat cw=(-c)*wni
    mat d=cw*g
    mat dt=trn(d)
    mat ddt=d*dt
    mat ddt=(c^(-1))*ddt
    mat a=ddt+wni
    mat wi=zer(n+1,n+1)

    for i=1 to n
      for j=1 to n
        let wi(i,j)=a(i,j)
        let zn1(j,1)=zp(j,1)
      next j
      let wi(n+1,i)=d(i,1)
      let wi(i,n+1)=d(i,1)
    next i

    let wi(n+1,n+1)=c
    mat bn1=wi*zn1
end sub


sub ftest(n,kf,stp,yd(),inda(),prev(),independent(,),bn1(,),f,msen1,ssen,ssen1,d$)
```

```
    dim yc(1,1)

 ! Ftest calculates the new model error (ssen1) and the F-statistic (f).


    let ssen1=0
    call model(yc,bn1,n,kf,stp,inda,independent,prev,d$)

    for k=1 to kf
       let ssen1=ssen1+(yc(1,k)-yd(k))^2
    next k

    let f=(ssen-ssen1)*(kf-n-1)/ssen1
    let msen1=ssen1/(kf-n-1)
 end sub



 sub ttest(n,msen1,bn1(,),wi(,),t())

 ! Ttest calculates the t-statistic for each variable in the model.


    for var=1 to n+1
       let t(var)=abs(bn1(var,1)/(msen1*wi(var,var)))
    next var
 end sub



 sub x1(kf,i,stp,ind,d$,xn1(,),independent(,))

 ! Fill the new independent data array, xn1, and invert the data if applicable.


    mat xn1=zer(kf,1)
    if ind<>8 then
      for k=1 to kf
        let xn1(k,1)=independent((k-i)*stp,ind)
      next k
    else
      for k=1 to kf
        let xn1(k,1)=1
      next k
    end if

 ! ** These may have to be changed if the independent variables are altered. **

    if d$="Teao" and (ind=4 or ind=1 or ind=9 or ind=11 or ind=12) then
      for k=1 to kf
        let xn1(k,1)=1/independent((k-i)*stp,ind)
      next k
    end if
    if d$="Pcond" and ind=4 then
```

```
    for k=1 to kf
      let xn1(k,1)=1/independent((k-i)*stp,ind)
    next k
  end if
  if d$="Pevap" and (ind=4 or ind=1 or ind=9 or ind=11 or ind=12) then
    for k=1 to kf
      let xn1(k,1)=1/independent((k-i)*stp,ind)
    next k
  end if
end sub


sub auto(n,kf,lag,stp,ind,inda(),no,ssen,msen,prev(),independent(,),yd(),
              xn(,),xn1(,),zn(,),zn1(,),bn(,),bn1(,),wi(,),wni(,),id$(),d$,indmax)

! Auto determines the most significant model variable and lag, or the most significant lag
! of a selected variable, depending on the independent variable selected (ind). The inverse
! of the standard deviation of error is used as the criteria when n=1, otherwise the
! F-statistic is used.


  dim fs(1),max(4),t(1),maxt(3),tot(1,1)
  mat t=zer(n+1)
  mat maxt=zer(3)
  mat tot=zer(indmax-1,3)
  let indt=ind

  if indt=indmax then
    let all=indmax-1
  else
    let all=1
  end if

  for var=1 to all
    let max(1)=-1e6
    if indt=indmax then
      let ind=var
      let inda(n+1)=var
      if var=7 then
        let prev(n+1)=1
      else
        let prev(n+1)=0
      end if
    end if
    if ind<>7 then
      let l=0
      let intr=-lag
    else
      let l=1
      let intr=-lag-1
    end if
    let intrt=intr
    do while intr>10
```

```
    let j=0
    let max(1)=0
    let rint=round(intr/10)
    mat fs=zer(round(intr/rint+.5))

    for i=l to l+intr step rint
      let j=j+1
       call x1(kf,i,stp,ind,d$,xn1,independent)
      if n<>0 then
        let prev(n+1)=i
         call winv(n,kf,stp,xn,xn1,yd,wni,wi,zn,zn1,bn1,bn,no)
         call ftest(n+1,kf,stp,yd,inda,prev,independent,bn1,f,msen1,ssen,ssen1,d$)
      else
          call first(kf,stp,i,ind,inda,xn1,independent,yd,zn,wni,bn,bn1,
                     ssen1,msen,msen1,d$)
        let f=1/sqr(abs(msen1))
      end if
      let fs(j)=f
      if f>max(1) then
        let max(1)=f
        let max(2)=j
        let max(3)=i
        let max(4)=msen1
      end if
    next i

    if max(2)>1 and max(2)<j-1 then
      let l=max(3)-rint+1
      let intr=2*rint-2
    else
      if rint>(-lag-max(3)) then
        let l=max(3)-rint+1
        let intr=rint+(-lag-max(3))-1
      else
        let l=max(3)
        let intr=rint-1
      end if
    end if
    print
    call indlabel(ind,d$,label$)
    print label$;
    print " test trend:"
    mat print fs
  loop

  let j=0
  mat fs=zer(round(intr+.5))
  for i=l to l+intr
    let j=j+1
     call x1(kf,i,stp,ind,d$,xn1,independent)
    if n<>0 then
      let prev(n+1)=i
       call winv(n,kf,stp,xn,xn1,yd,wni,wi,zn,zn1,bn1,bn,no)
       call ftest(n+1,kf,stp,yd,inda,prev,independent,bn1,f,msen1,ssen,ssen1,d$)
```

```
        else
            call first(kf,stp,i,ind,inda,xn1,independent,yd,zn,wni,bn,bn1,ssen1,
                       msen,msen1,d$)
          let f=1/sqr(abs(msen1))
        end if
        let fs(j)=f
        if f>max(1) then
          let max(1)=f
          let max(3)=i
          let max(4)=msen1
        end if
      next i

      if intrt<10 then
        print
        call indlabel(ind,d$,label$)
        print label$;
        print " test trend:"
        mat print fs
      end if
      if indt=indmax then
        let tot(var,1)=max(1)
        let tot(var,2)=max(3)
        let tot(var,3)=max(4)
        if max(1)>maxt(1) then
          let maxt(1)=max(1)
          let maxt(2)=max(3)
          let maxt(3)=var
        end if
        let max(3)=maxt(2)
      end if
    next var
    if indt=indmax then
      let ind=maxt(3)
      let inda(n+1)=ind
      call autodisp(n,ind,tot,d$,id$)
    end if
    let prev(n+1)=max(3)
end sub



sub autodisp(n,ind,tot(,),d$,id$())

! ** Autodisp outputs the results of the Auto subroutine to the screen. **


    let form$="#########     ###     ########.##     ####.#"
    print
    print "Maximum criteria values for each independent variable:"
    print
    print " Variable      Lag       ";
    if n<>0 then
      print "   F         ";
```

```
    else
      print "  1/std       ";
    end if
    print "Std"
    print
    if d$="Pevap" or d$="Teao" then
       print using form$:"1/Ncomp",tot(1,2),tot(1,1),sqr(abs(tot(1,3)))
       print using form$:"1/Nc^2",tot(9,2),tot(9,1),sqr(abs(tot(9,3)))
    else
       print using form$:"Ncomp",tot(1,2),tot(1,1),sqr(abs(tot(1,3)))
       print using form$:"Nc^2",tot(9,2),tot(9,1),sqr(abs(tot(9,3)))
    end if
    print using form$:"Tcai",tot(2,2),tot(2,1),sqr(abs(tot(2,3)))
    print using form$:"Teai",tot(3,2),tot(3,1),sqr(abs(tot(3,3)))
    print using form$:"Teai^2",tot(13,2),tot(13,1),sqr(abs(tot(13,3)))
    print using form$:"Te/Nc",tot(10,2),tot(10,1),sqr(abs(tot(10,3)))
    if d$="Teao" or d$="Pcond" or d$="Pevap" then
       print using form$:"1/Vdac",tot(4,2),tot(4,1),sqr(abs(tot(4,3)))
       print using form$:"1/Vdac^2",tot(12,2),tot(12,1),sqr(abs(tot(12,3)))
       print using form$:"Nc/Vdac",tot(11,2),tot(11,1),sqr(abs(tot(11,3)))
    else
       print using form$:"Vdac",tot(4,2),tot(4,1),sqr(abs(tot(4,3)))
       print using form$:"Vdac^2",tot(12,2),tot(12,1),sqr(abs(tot(12,3)))
       print using form$:"Vdac/Nc",tot(11,2),tot(11,1),sqr(abs(tot(11,3)))
    end if
    print using form$:"Vdae",tot(5,2),tot(5,1),sqr(abs(tot(5,3)))
    print using form$:"Constant",tot(8,2),tot(8,1),sqr(abs(tot(8,3)))
    print using form$:d$,tot(7,2),tot(7,1),sqr(abs(tot(7,3)))
    print
    call indlabel(ind,d$,id$(n+1))
end sub


sub varlist

! ** Varlist prints the independent and dependent variable names which are available. **


   print
   print
   print "    Dependent Variables","  Independent Variables"
   let form$="         #########                  ###########"
   print
   print using form$:"Teao","Ncomp"
   print using form$:"Power","Tcai"
   print using form$:"Pcond","Teai"
   print using form$:"Pevap","Vdac"
   print using form$:"  ","Vdae"
   print using form$:"  ","Rhi"
   print using form$:"  ","Nc^2"
   print using form$:"  ","Teai^2"
   print using form$:"  ","Teai/Nc"
   print using form$:"  ","Vdac/Nc"
```

```
    print using form$:"  ","Vdac^2"
    print using form$:"  ","Constant"
    print using form$:"  ","Dependent"
    print using form$:"  ","Auto"
    print using form$:"  ","Model check"
    print
    print
  end sub


 sub dselect(kf,lag,stp,d$,dependent(,),independent(,),yd())

 ! ** Dselect is the user interface which selects the desired dependent variable. The routine
 ! also loads the dependent data into the proper column in the independent array. **


    let dep=0
    mat yd=zer(lag+1:kf)

    do while dep=0
       input prompt "Select dependent variable: ":dvar$
       let dvar$=lcase$(dvar$)
       if dvar$[1:1]="t" then let dep=3
       if dvar$[1:2]="po" then let dep=4
       if dvar$[1:2]="pc" then let dep=1
       if dvar$[1:2]="pe" then let dep=2
       if dep=0 then print "Not a valid variable"
    loop

    if dep=1 then let d$="Pcond"
    if dep=2 then let d$="Pevap"
    if dep=3 then let d$="Teao"
    if dep=4 then let d$="Power"

    for k=lag+1 to kf
       let yd(k)=dependent(k*stp,dep)
       let independent(k*stp,7)=yd(k)
    next k

    print
 end sub


 sub indselect(n,kf,lag,stp,id$(),d$,prev(),ind,inda(),independent(,),bn(,),bn1(,),xn(,),
                 xn1(,),yd(),no,newind$,auto$,indmax,man)

 ! ** Indselect is the user interface which selects the independent variable (or Auto), and the
 ! lag for that variable (or Auto). Some bookkeeping is also done here. **


    dim tempx(1,1),tempi(1),tempb(1,1),yc(1,1),bnt(1,1),xnt(1,1)
    let man$="y"
```

```
if man<>1 then
  if n=0 then
    mat xn=zer(kf,1)
    mat bn1=zer(1,1)
    mat bn=zer(1,1)
  end if

  if n<>0 and no<>n then
    mat tempx=zer(kf,n-1)
    mat tempx=xn
    mat xn=zer(kf,n)

    for k=1 to kf
      for i=1 to n-1
        let xn(k,i)=tempx(k,i)
      next i
      let xn(k,n)=xn1(k,1)
    next k
  end if
end if

mat xn1=zer(kf,1)
let man=0

do while man$="y"
  let ind=0

  do while ind=0
    input prompt "Select independent variable (or None): ":indvar$
    let indvar$=lcase$(indvar$)
    if indvar$[1:1]="n" then let ind=99
    if indvar$[1:1]="d" then let ind=7
    if indvar$[1:2]="nc" then let ind=1
    if indvar$[1:3]="nc2" then let ind=9
    if indvar$[1:2]="tc" then let ind=2
    if indvar$[1:4]="vdac" then let ind=4
    if indvar$[1:3]="vdn" then let ind=11
    if indvar$[1:4]="vdc2" then let ind=12
    if indvar$[1:4]="vdae" then let ind=5
    if indvar$[1:2]="te" then let ind=3
    if indvar$[1:3]="te2" then let ind=13
    if indvar$[1:3]="ten" then let ind=10
    if indvar$[1:1]="r" then let ind=6
    if indvar$[1:1]="a" then let ind=indmax
    if indvar$[1:1]="c" then let ind=8
    if indvar$[1:1]="m" then
      let man=1
      let n=0
      print
      print
      print "Enter model:"
      print
    else
```

```
      if ind=0 then
        print "Not a valid variable"
      end if
    end if
  loop

  if man<>1 or ind=99 then let man$="n"

  if ind<>99 then
    mat tempi=inda
    mat inda=zer(n+1)
    for i=1 to n
      let inda(i)=tempi(i)
    next i
    let inda(n+1)=ind
  end if

  if ind<>99 and ind<>indmax then
    let prev(n+1)=1e6


    do while prev(n+1)>-lag
      if ind<>7 and ind<>8 then
        print "Select lag ( 0 -";-lag;
        input prompt ") or Auto: ":in$
      else
        if ind=7 then
          print "Select lag ( 1 -";-lag;
          input prompt ") or Auto: ":in$
        end if
      end if
      let in$=lcase$(in$)
      if ord(in$[1:1])>47 and ord(in$[1:1])<58 then
        let prev(n+1)=val(in$)
        let prev(n+1)=abs(prev(n+1))
        let auto$="n"
      else
        if ind<>8 then
          let auto$="y"
          let prev(n+1)=1
        else
          let auto$="n"
          let prev(n+1)=0
        end if
      end if
    loop

  end if


  if ind<>99 then
    call indlabel(ind,d$,id$(n+1))
    if ind=indmax then let auto$="y"
    if man=1 then
```

```
        mat tempb=bn1
        mat bn1=zer(n+1,1)
        for i= 1 to n
          let bn1(i,1)=tempb(i,1)
        next i
        print
        input prompt "Coefficient = ":bn1(n+1,1)
        print
      end if
    else
      if man<>1 then let newind$="n"
    end if
    if man$="y" then let n=n+1
  loop

end sub


sub indlabel(ind,d$,label$)

! ** Indlabel assigns the variable a name. **


  if ind=1 then let label$="Ncomp"
  if ind=2 then let label$="Tcai"
  if ind=3 then let label$="Teai"
  if ind=13 then let label$="Te^2"
  if ind=4 then let label$="Vdac"
  if ind=5 then let label$="Vdae"
  if ind=6 then let label$="Rhi"
  if ind=7 then let label$=d$
  if ind=8 then let label$="Const"
  if ind=9 then let label$="Nc^2"
  if ind=10 then let label$="Te/Nc"
  if ind=11 then let label$="Vdac/Nc"
  if ind=12 then let label$="Vdac^2"
  if d$="Teao" and (ind=4 or ind=1 or ind=9 or ind=11 or ind=12) then let
                    label$="1/"&label$
  if d$="Pcond" and (ind=4 or ind=11) then let label$="1/"&label$
  if d$="Pevap" and (ind=1 or ind=4 or ind=9 or ind=11 or ind=12) then let
                    label$="1/"&label$
end sub


sub indat(kf,lag,dependent(,),independent(,),snd$,indmax)

! ** Indat reads the data file specified by the user. The data file contains the total number
! of data points (kf), the maximum number of lag data available (lag), and a row of column
! labels. The data is read in to the appropriate array, and manipulations are performed. **


  dim temp(1,54),label$(54)              !** These dimension are for the columns (54)
```

```
!                                                    in the data file. **

   input prompt "Data file? ":data$
   let data$=data$&".table"
   open #1:name data$, access input, org text
   input #1: kf
   input #1: lag
   print "Number of time steps (<";kf;
   input prompt ")? ":kf
   print "Number of previous steps available (<";lag;
   input prompt ")? ":lag
   let kf=kf-lag
   let lag=-lag
   print
   print "          Processing data..."
   mat dependent=zer(lag+1:kf,4)
   mat independent=zer(lag+1:kf,indmax-1)
   mat input #1:label$
   print tab(12,20); "Minutes Remaining:"

   for k=lag+1 to kf
      if k=lag+1 then let t1=time
      mat input #1:temp
      let independent(k,1)=temp(1,24)                  !Ncomp
      let independent(k,2)=temp(1,2)                   !Tcai
      let independent(k,3)=temp(1,10)                  !Teai
      let independent(k,4)=temp(1,4)                   !Vdac
      let independent(k,5)=temp(1,14)                  !Vdae
      let independent(k,6)=temp(1,11)                  !Rhei
      let independent(k,8)=1                           !Constant
      let independent(k,9)=temp(1,24)^2                !Nc^2
      let independent(k,10)=temp(1,10)/temp(1,24)      !Teai/Nc
      let independent(k,13)=temp(1,10)^2               !Teai^2
      let independent(k,11)=temp(1,4)/temp(1,24)       !Vdac/Nc
      let independent(k,12)=temp(1,4)^2                !Vdac^2


      let dependent(k,1)=temp(1,6)                     !Pci
      let dependent(k,2)=temp(1,18)                    !Peo
      let dependent(k,3)=temp(1,12)                    !Teao
      let dependent(k,4)=temp(1,44)                    !Power

      if k=lag+1 then let t2=time
      if k=lag+1 or k/10=int(k/10) then
        print tab(12,38);"          "
        print tab(12,38); round((kf-k)*(t2-t1)/60,2)
      end if
   next k

   close #1
   if snd$="y" then call beep
end sub
```

```
sub beep
   sound 750,.075
end sub


sub result(n,kf,lag,stp,id$(),prev(),d$,bn(,),bn1(,),f,t(),msen,msen1,ssen,ssen1,
                ft,tt(),msent,msen1t,bnt(,),bnte(,),snd$,xn(,),xn1(,),yd(),independent(,),
                ind,inda(),man)

! Result prints out the current and previous models, the t-statistic, the number of
! parameters, the standard deviation of error, and the F-statistic. The graph may also be
! viewed. The user is asked whether the new variable should be kept in the model.


  dim xtp(1,1)
  let form$=" ###########    ###   -#.####^^^  -#.####^^^    -#.##^^^"
  let form1$=" ###########    ###         -#.####^^^"
  print
  print "Dependent Variable: ";d$
  print
  print "Step:";stp
  print
  print "                        Coefficients"

  if man<>1 then
    print " Variable      Lag      Old       New       t"
  else
    print " Variable      Lag"
  end if

  print

  if n<>0 and man<>1 then
     for i=1 to n
        print using form$: id$(i),prev(i),bn(i,1),bn1(i,1),t(i)
     next i
  else
     if man=1 then
       for i=1 to n
          print using form1$: id$(i),prev(i),bn1(i,1)
       next i
     end if
  end if
  if man<>1 then
    print using form$: id$(n+1),prev(n+1),"-----",bn1(n+1,1),t(n+1)
  else
    print using form1$: id$(n+1),prev(n+1),bn1(n+1,1)
  end if

  print
  print
  print " n(old) = ";n
```

```
    print " n(new) = ";n+1

    if n<>0 and man<>1 then
       print "      F = ";f
       print "  SD(n) = ";sqr(msen)
    end if

    print "SD(n+1) = ";sqr(msen1)
    print "Degrees of Freedom: ";kf-n-1
    print
    if snd$="y" then call beep
    input prompt "See graph? ":ans$
    let ans$=lcase$(ans$[1:1])
    if ans$="y" or ans$="m" then call graph(n,kf,stp,lag,prev,bn,bn1,xn,xn1,yd,d$,ans$,
                                                    inda,independent,man)
    let n=n+1
    if man<>1 then
      input prompt "Retain this variable? ":ans$
      let ans$=lcase$(ans$[1:1])
      if ans$="y" then
        let ft=f
        let ssen=ssen1
        let msent=msen
        let msen=msen1
        let msen1t=msen1
        mat tt=t
        mat bnt=bn
        mat bn=bn1
        mat bnte=bn1
      else
        let f=ft
        let n=n-1
        if n<>0 then mat t=tt
        mat bn1=bnte
      end if
    end if
end sub



sub keep(n,kf,stp,lag,id$(),prev(),d$,bn(,),bn1(,),f,t(),msen,msen1,yd(),
                xn(,),independent(,),inda(),man)

! Keep allows the model data to be stored on disk.  The model parameters, the calculated
! model values and the data (as comma-delimited text), or both may be saved.


    dim bn1t(1,1),yc(1,1),xnt(1,1)
    let form$=" ###########    ###    -#.####^^^   -#.####^^^     -#.##^^^"
    let form1$=" ###########    ###          -#.####^^^"
    print
    input prompt "Save results? ":save$
    let save$=lcase$(save$)
    if save$="y" then
```

```
input prompt "Save Coefficients or Fit data (or Both)? ":type$
let type$=lcase$(type$[1:1])
input prompt "Name of file: ":file$
if type$="c" or type$="b" then
   open #2:name file$&".coef", create "newold"
   erase #2
   print #2:
   print #2: "Dependent Variable: ";d$
   print #2:
   print #2: "Step:";stp
   print #2:
   print #2: "                    Coefficients"
   if man<>1 then
     print #2: " Variable      Lag      Old      New      t"
   else
     print #2: " Variable      Lag"
   end if

   print

   if n<>0 and man<>1 then
      for i=1 to n-1
         print #2, using form$: id$(i),prev(i),bn(i,1),bn1(i,1),t(i)
      next i
   else
      if man=1 then
        for i=1 to n-1
           print #2, using form1$: id$(i),prev(i),bn1(i,1)
        next i
      end if
   end if
   if man<>1 then
      print #2, using form$: id$(n),prev(n),"-----",bn1(n,1),t(n)
   else
     print #2, using form1$: id$(n),prev(n),bn1(n,1)
   end if

   print #2:
   print #2:
   print #2: " n(old) = ";n-1
   print #2: " n(new) = ";n

   if n>1 and man<>1 then
      print #2: "     F = ";f
      print #2: "  SD(n) = ";sqr(msen)
   end if

   print #2: "SD(n+1) = ";sqr(msen1)
   print #2: "Degrees of Freedom: ";kf-n
   print #2:
   close #2
end if
if type$="f" or type$="b" then
   open #2:name file$&".fit", create "newold"
```

```
        erase #2

        if d$[1:1]="P" then let unit$="  (psig)"
        if d$[1:2]="Po" then let unit$="  (Btu/hr)"
        if d$[1:1]="S" or d$[1:1]="T"  then let unit$="  (°F)"
        if d$[1:1]="R" then let unit$="  (%)"

        print #2:"Time  (sec),";d$&unit$;",Predicted,Ncomp,Tcai,Teai,Vdac,Vdae,Rhei"

        call model(yc,bn1,n,kf,stp,inda,independent,prev,d$)
        for k=lag+1 to 0

! ** The specific columns in independent may have to be changed **

          print #2: k*stp;",";yd(k);",";",";independent(k*stp,1);",";
          print #2: independent(k*stp,2);",";independent(k*stp,3);",";
                    independent(k*stp,4);",";
          print #2: independent(k*stp,5);",";independent(k*stp,6)
        next k
        for k=1 to kf
          print #2: k*stp;",";yd(k);",";yc(1,k);",";independent(k*stp,1);",";
          print #2: independent(k*stp,2);",";independent(k*stp,3);",";
                    independent(k*stp,4);",";
          print #2: independent(k*stp,5);",";independent(k*stp,6)
        next k

        close #2
      end if
    end if
end sub



sub graph(n,kf,stp,lag,prev(),bn(,),bn1(,),xn(,),xn1(,),yd(),d$,ans$,inda(),
                independent(,),man)

! Graph plots the current model values, the previous model values, and the dependent data.
! The maximum value of the x and y-axis are determined automatically, while the y-axis
! minimum can either be the calculated minimum value or the default, 0, depending on the
! value of ans$ ("m" is min and max).


    dim bnt(1,1),bn1t(1,1),yc(1,1),yb(1,1),xnt(1,1),xn1t(1,1),vplot(1,2),tempx(1,1)
   mat yb=zer(1,kf)
   let tme=kf*stp
   open #3: screen .55,1,.02,.52
   ask pixels px,py
   call model(yc,bn1,n+1,kf,stp,inda,independent,prev,d$)
   if man<>1 then call model(yb,bn,n,kf,stp,inda,independent,prev,d$)
   let maxind=0
   let minind=yd(1)
   for k=1 to kf
      if yc(1,k)>maxind then let maxind=yc(1,k)
      if yd(k)>maxind then let maxind=yd(k)
```

```
    if yc(1,k)<minind then let minind=yc(1,k)
    if yd(k)<minind then let minind=yd(k)
 next k
 if maxind=int(maxind) then let maxind=maxind+.1
 let morder=pos(str$(maxind),".")-2
 let maxind=round(maxind+.5*10^morder,-morder)
 if ans$="m" then
   let minorder=pos(str$(minind),".")-2
   let minind=round(minind-.5*10^minorder,-minorder)
   if (maxind-minind)/(10^(morder))<2 then let morder=morder-1
 else
   let minind=0
 end if
 let order=len(str$(tme))-1
 let tme=round(tme+.5*10^order,-order)
 set window 0,tme,minind,maxind
 plot lines:0,maxind;0,minind;tme,minind
 let max=(maxind-minind)/(10^(morder))
 for i=1 to max
    plot lines: 0,i*(10^(morder))+minind;.01*tme,i*(10^(morder))+minind
 next i
 let max=tme/(10^(order))
 for i=1 to max
    plot lines: i*(10^(order)),minind;i*(10^(order)),.01*(maxind-minind)+minind
 next i
 mat vplot=zer(kf,2)
 for k=1 to kf
    let vplot(k,1)=k*stp
    let vplot(k,2)=yd(k)
 next k
 plot text, at .8*tme,(.0625+.025*426/py)*(maxind-minind)+minind : d$
 mat plot lines: vplot
 if n<>0 then
   for k=1 to kf
      let vplot(k,2)=yb(1,k)
   next k
   set color "red"
   mat plot lines: vplot
   plot text, at .8*tme,.0625*(maxind-minind)+minind : "Previous"
 end if
 for k=1 to kf
    let vplot(k,2)=yc(1,k)
 next k
 set color "blue"
 if n<>0 then
   plot text, at .8*tme,(.0625-.025*426/py)*(maxind-minind)+minind : "New"
 else
   plot text, at .8*tme,.0625*(maxind-minind)+minind : "New"
 end if
 mat plot lines: vplot
 set color "black"
 plot text, at .5*tme,.975*(maxind-minind)+minind : d$
 plot text, at .1*tme,(.0625+.0125*426/py)*(maxind-minind)+minind : "x: 0 -
                        "&str$(tme)
```

```
    plot text, at .1*tme,(.0625-.0125*426/py)*(maxind-minind)+minind : "y:
                          "&str$(minind)&" - "&str$(maxind)
    get key z
    clear
    close #3
  end sub


 sub model(yc(,),bn1(,),n,kf,stp,inda(),independent(,),prev(),d$)

 ! Model calculates the model values given the model coefficients and the independent data.


    dim bnt(1,1),indp(1,1),temp(1,1)

    mat indp=zer(kf,n)
    mat yc=zer(1,kf)

    for i=1 to n
         call x1(kf,prev(i),stp,inda(i),d$,temp,independent)
         for k=1 to kf
            let indp(k,i)=temp(k,1)
         next k
    next i

    for k=1 to kf
       for i=1 to n
          if inda(i)<>7 then
             let yc(1,k)=yc(1,k)+bn1(i,1)*indp(k,i)
          else
            if k>prev(i) then
               let yc(1,k)=yc(1,k)+bn1(i,1)*yc(1,(k-prev(i)))
            else
               let yc(1,k)=yc(1,k)+bn1(i,1)*independent((k-prev(i))*stp,7)
            end if
          end if
       next i
    next k

 end sub
```

APPENDIX B

Data Reduction Program

TrueBasic™ Version 2.02

Developed on MacIntosh IIci

```
! THIS PROGRAM WILL CALCULATE REFRIGERANT AND AIR PROPERTIES.
! THE PROGRAM WILL CALCULATE THE PROPERTIES AS AN AVERAGE OF
! THE DATA, OR INDEPENDENTLY. THE DATA MAY BE SAVED IN TABLE
! FORM OR AS COMMA-DELIMITED TEXT. Version 2c
! Charles Schenk
! University of Illinois

DIM DATA(17,17),AVGDATA(1,17),Temp(1,27)
DECLARE DEF entrop,sten,rhof
DECLARE DEF Hvap,Hliq,Vol,T,CPliq,CPgas
DECLARE DEF Mdvap,hSHV,sSHV,Mdliq,Mddisch
let view$="y"

! OPEN INPUT FILE AND OUTPUT FILES

print "Data set to be analyzed";
input data$
let data$=ucase$(data$)
OPEN #1: NAME data$ ,ACCESS INPUT, ORG TEXT

input #1: series$
if series$="s" then input #1: intr
INPUT #1: SETS
INPUT #1: Prev

print "Save output (y)";
input save$
let save$=lcase$(save$)
if save$="y" then
  print "Name of file";
  input file$
  let file$=ucase$(file$)
  print "Type of file (Table or Normal)";
  input type$
  let type$=lcase$(type$[1:1])

  do while type$<>"n" and type$<>"t"
    print "Type T or N";
    input type$
    let type$=lcase$(type$[1:1])
  loop
```

```
    if type$="t" then
      open #3: name file$&".table",create "newold"
      ask #3: filesize fs
      if fs<>0 then
        print "Add to existing file (y)";
        input add$
        let add$=lcase$(add$)
        if add$<>"y" then
          close #3
          open #3: name file$&".table 1",create "newold"
        end if
      end if
    else
      OPEN #2: NAME file$&".norm",CREATE "NEWOLD"
      ERASE #2
    end if
 print "View output (y)";
 input view$
 let view$=lcase$(view$)
 end if

 let tme=0

 FOR XXX = 1 TO SETS STEP 1


 print
 if series$="s" and XXX=1 then
   INPUT #1:title$
 end if
 if series$="a" then
   INPUT #1:title$
 end if

 if series$="s" then
    if XXX=1 then
      clear
      print tab(7,5); "ANALYZING DATA SET: ";title$;"  Time = "
      let t1=time
    end if
    print tab(7,34+len(title$));tme
 else
    print  "ANALYZING DATA SET: ";title$
 end if


 if series$="s" then
   let NTEST=1
 else
   INPUT #1:NTEST
 end if

 ! REDIMENSION ARRAYS
```

```
MAT REDIM DATA(NTEST,27)
MAT AVGDATA = ZER(1,27)


! READ 1 SET OF DATA FROM INPUT FILE

if series$="s" and XXX=1 then
  INPUT #1:PAMBI                                  ! AMBIENT PRESSURE, PSIA
  let PAMBI=PAMBI*.4912
end if
if series$="a" then
  INPUT #1:PAMBI                                  ! AMBIENT PRESSURE, PSIA
  let PAMBI=PAMBI*.4912
end if

if series$="a" then

  For i=1 to NTEST                                          ! SYSTEM DATA
    mat input #1:temp
    for j=1 to 27
      let DATA(i,j)=temp(1,j)
    next j
  next i

! AVERAGE ALL SYSTEM DATA

  FOR X = 1 TO 27 STEP 1
    LET SUM = 0.0
      FOR Y = 1 TO NTEST STEP 1
        LET SUM = DATA(Y,X) + SUM
      NEXT Y
    LET AVGDATA(1,X) = SUM/NTEST
  NEXT X
else
  mat input #1:temp
  mat avgdata=temp
end if

let TWBI=avgdata(1,27)
let avgdata(1,3)=4794.5539*avgdata(1,3)^.73637923           ! COND SCFM
let avgdata(1,12)=262.920053609*avgdata(1,12)^.57142857     ! EVAP SCFM

! CONVERT DATA TO SI UNITS

LET PAMB  = PAMBI*6.89478                                   ! KPA
LET TDB   = (AVGDATA(1,1)-32)/1.8                           ! C
LET TWB   = (TWBI-32)/1.8                                   ! C
CALL WAMB (PAMB,TDB,TWB,WIN)                                ! HUMIDITY
LET CATI  = (AVGDATA(1,1)-32)/1.8+273.15                    ! K
LET CATO  = (AVGDATA(1,2)-32)/1.8+273.15                    ! K

LET CAHI = 1.005*(CATI-273.15)+WIN*(2501+1.805*(CATI-273.15))
                                          ! AIR ENTHALPY IN, W OUT
```

```
LET CAFLO = (AVGDATA(1,3)*27.2155*.0764)
                                        ! COND AIR MASS FLOW, KG/HR

LET CRTI  = (AVGDATA(1,4)-32)/1.8+273.15                    ! K
LET CRPI  = (AVGDATA(1,5)*6.89478)+PAMB                     ! KPA
LET CRTO  = (AVGDATA(1,6)-32)/1.8+273.15                    ! K
LET CRPO  = (AVGDATA(1,7)*6.89478)+PAMB                     ! KPA
LET EATI  = (AVGDATA(1,8)-32)/1.8+273.15                    ! K
LET EATO  = (AVGDATA(1,10)-32)/1.8+273.15                   ! K
LET EARHI = (AVGDATA(1,9))                                  ! RH
LET EARHO = (AVGDATA(1,11))                                 ! RH

CALL MOISTH(PAMB,EARHO,EATO,EAHO,EAWO)    ! AIR ENTHALPY OUT
LET EAFLO = (AVGDATA(1,12)*27.2155*.0764) ! EVAP AIR MASS FLOW, KG/HR
LET ERTI  = (AVGDATA(1,13)-32)/1.8+273.15                   ! K
LET ERPI  = (AVGDATA(1,14)*6.89478)+PAMB                    ! KPA
LET ERTO  = (AVGDATA(1,15)-32)/1.8+273.15                   ! K
LET ERPO  = (AVGDATA(1,16)*6.89478)+PAMB                    ! KPA
LET KRTI  = (AVGDATA(1,17)-32)/1.8+273.15                   ! K
LET KRPI  = (AVGDATA(1,18)*6.89478)+PAMB                    ! KPA
LET KRTO  = (AVGDATA(1,19)-32)/1.8+273.15                   ! K
LET KRPO  = (AVGDATA(1,20)*6.89478)+PAMB                    ! KPA
if avgdata(1,22)<10 then let avgdata(1,21)=0          ! Clutch engaged?
LET KRPM  = (AVGDATA(1,21))                                 ! RPM
LET KTRQ  = (AVGDATA(1,22))                                 ! INLB
let Msuc  = Mdvap(ERPO,ERTO,avgdata(1,23))                  ! lb/hr
let Mdis  = Mddisch(KRPO,KRTO,avgdata(1,25))                ! lb/hr
let Mliq  = Mdliq(CRTO,avgdata(1,24))                       ! lb/hr
LET RFLOW = (Mliq*.4536)                                    ! KG/HR


! CALCULATE ALL REFRIGERANT AND REMAINING AIR PROPERTIES

CALL MOISTH(PAMB,CARHI,CATI,CAHI,CAWI)
LET  CAHO = 1.005*(CATO-273.15) + CAWI*(2501+1.805*(CATO-273.15))
LET  CRHI = hSHV(CRTI-T(CRPI),T(CRPI))
LET  CRHO = Hliq(CRTO)
LET  CRFLOW = CAFLO*(CAHO-CAHI)/(CRHI-CRHO)
CALL MOISTH(PAMB,EARHI,EATI,EAHI,EAWI)
let  dwater=EAFLO*(EAWI-EAWO)
LET  ERHO = hSHV(ERTO-T(ERPO),T(ERPO))
let  ERHI = CRHO
let  KRHI = hSHV(KRTI-T(KRPI),T(KRPI))
let  KRHO = hSHV(KRTO-T(KRPO),T(KRPO))


! CALCULATE CAPACITIES, COP, ETC.

let Power=AVGDATA(1,21)*AVGDATA(1,22)/24.769
if power=0 then let power=0.01
let Capacity=(ERHO-ERHI)*0.42992*Mliq
let Qcomp=Power-(KRHO-KRHI)*0.42992*Mliq
let Qcond=(CRHI-CRHO)*0.42992*Mliq
```

```
let Qsuct=(KRHI-ERHO)*0.42992*Mliq
let Qdisc=(KRHO-CRHI)*0.42992*Mliq
let Esuper=ERTO-T(ERPO)
let Csub=T(CRPO)-CRTO
let Tcsat=T(CRPO)


! PRINT OUTPUT (ALL PROPERTIES AND MEASUREMENTS)

LET FORM$ = "################## ######.# <######"
if view$="y" then
 print " "
 if series$="s" then
   print "  Time =";tme
 else
   print
 end if
 PRINT " "
 PRINT using form$:  "COND AIR T IN   =",AVGDATA(1,1),"°F"
 PRINT using form$:  "COND AIR T OUT  =",AVGDATA(1,2),"°F"
 PRINT using form$:  "COND AIR CFM    =",AVGDATA(1,3),"SCFM"
 PRINT using form$:  "COND REF T IN   =",AVGDATA(1,4),"°F"
 PRINT using form$:  "COND REF Pg IN  =",AVGDATA(1,5),"psig"
 PRINT using form$:  "COND REF T OUT  =",AVGDATA(1,6),"°F"
 PRINT using form$:  "COND REF Pg OUT =",AVGDATA(1,7),"psig"
 print using form$:  "Cond Ref dp     =",AVGDATA(1,5)-AVGDATA(1,7),"psid"
 PRINT using form$:  "EVAP AIR T IN   =",AVGDATA(1,8),"°F"
 PRINT using form$:  "EVAP AIR RH IN  =",AVGDATA(1,9),"%"
 PRINT using form$:  "EVAP AIR T OUT  =",AVGDATA(1,10),"°F"
 PRINT using form$:  "EVAP AIR RH OUT =",AVGDATA(1,11),"%"
 PRINT using form$:  "EVAP AIR FLOW   =",AVGDATA(1,12),"SCFM"
 PRINT using form$:  "EVAP REF T IN   =",AVGDATA(1,13),"°F"
 PRINT using form$:  "EVAP REF Pg IN  =",AVGDATA(1,14),"psig"
 PRINT using form$:  "EVAP REF T OUT  =",AVGDATA(1,15),"°F"
 PRINT using form$:  "EVAP REF Pg OUT =",AVGDATA(1,16),"psig"
 print using form$:  "Evap Ref dp     =",AVGDATA(1,14)-AVGDATA(1,16),"psid"
 PRINT using form$:  "COMP REF T  IN  =",AVGDATA(1,17),"°F"
 PRINT using form$:  "COMP REF Pg IN  =",AVGDATA(1,18),"psig"
 PRINT using form$:  "COMP REF T OUT  =",AVGDATA(1,19),"°F"
 PRINT using form$:  "COMP REF Pg OUT =",AVGDATA(1,20),"psig"
 PRINT using form$:  "COMP SPEED      =",AVGDATA(1,21),"RPM"
 PRINT using form$:  "COMP TORQUE     =",AVGDATA(1,22),"in•lb"
 PRINT using form$:  "AMB AIR PRESS   =",PAMB/6.89478,"psia"
 PRINT using form$:  "AMB AIR T DB    =",TDB*1.8 + 32,"°F"
 PRINT using form$:  "AMB AIR T WB    =",TWB*1.8 + 32,"°F"
 PRINT using form$:  "COND AIR DELT H =",(CAHO-CAHI)*0.42992,"Btu/lb"
 PRINT using form$:  "COND AIR MFLOW  =",CAFLO/.4536,"lb/hr"
 PRINT using form$:  "COND REF H IN   =",CRHI*0.42992,"Btu/lb"
 PRINT using form$:  "COND REF H OUT  =",CRHO*0.42992,"Btu/lb"
 PRINT using form$:  "EVAP AIR DELT H =",(EAHI-EAHO)*0.42992,"Btu/lb"
 PRINT using form$:  "EVAP AIR MFLOW  =",EAFLO/.4536,"lb/hr"
 PRINT using form$:  "EVAP REF H IN   =",ERHI*0.42992,"Btu/lb"
 PRINT using form$:  "EVAP REF H OUT  =",ERHO*0.42992,"Btu/lb"
 PRINT using form$:  "COMP REF H IN   =",KRHI*0.42992,"Btu/lb"
```

```
  PRINT using form$:  "COMP REF H OUT  =",KRHO*0.42992,"Btu/lb"
  PRINT  " "
  PRINT using form$:  "SUCT FLOW RATE  =",Msuc,"lb/hr"
  print using form$:  "Disch. Flow Rate =",Mdis,"lb/hr"
  print using form$:  "Liq. Flow Rate  =",Mliq,"lb/hr"
  PRINT using form$:  "CALCULATED FLOW  =",CRFLOW/.4536,"lb/hr"
  PRINT  " "
  print using form$:  "COP       =",Capacity/Power
  print using form$:  "Power     =",Power,"Btu/hr"
  print using form$:  "Capacity  =",Capacity,"Btu/hr"
  print using form$:  "Qcomp     =",Qcomp,"Btu/hr"
  print using form$:  "Qcond     =",Qcond,"Btu/hr"
  print using form$:  "Qsuction  =",Qsuct,"Btu/hr"
  print using form$:  "Qdischarge =",Qdisc,"Btu/hr"
  print  " "
  print using form$:  "Evap. Superheat  =",Esuper*1.8,"°F"
  print using form$:  "Cond. Subcooling =",Csub*1.8,"°F"
  print  " "
  print using form$:  "Condensation Rate =",dwater/.4536,"lb/hr"
  print  " "
  print  " "

end if


! SAVE OUTPUT

if save$="y" then

  if type$="n" then
   print #2: " "
   PRINT #2: "   DATA SET ";title$;
   if series$="s" then
     print #2: "  Time =";tme
   else
     print #2:
   end if
   PRINT #2: " "
   PRINT #2: " "
   PRINT #2, using form$: "COND AIR T IN  =",AVGDATA(1,1),"°F"
   PRINT #2, using form$: "COND AIR T OUT  =",AVGDATA(1,2),"°F"
   PRINT #2, using form$: "COND AIR CFM   =",AVGDATA(1,3),"SCFM"
   PRINT #2, using form$: "COND REF T IN  =",AVGDATA(1,4),"°F"
   PRINT #2, using form$: "COND REF Pg IN  =",AVGDATA(1,5),"psig"
   PRINT #2, using form$: "COND REF T OUT  =",AVGDATA(1,6),"°F"
   PRINT #2, using form$: "COND REF Pg OUT =",AVGDATA(1,7),"psig"
   print #2, using form$: "Cond Ref dp    =",AVGDATA(1,5)-AVGDATA(1,7),"psid"
   PRINT #2, using form$: "EVAP AIR T IN  =",AVGDATA(1,8),"°F"
   PRINT #2, using form$: "EVAP AIR RH IN  =",AVGDATA(1,9),"%"
   PRINT #2, using form$: "EVAP AIR T OUT  =",AVGDATA(1,10),"°F"
   PRINT #2, using form$: "EVAP AIR RH OUT =",AVGDATA(1,11),"%"
   PRINT #2, using form$: "EVAP AIR FLOW  =",AVGDATA(1,12),"SCFM"
   PRINT #2, using form$: "EVAP REF T IN  =",AVGDATA(1,13),"°F"
   PRINT #2, using form$: "EVAP REF Pg IN  =",AVGDATA(1,14),"psig"
```

```
    PRINT #2, using form$: "EVAP REF T OUT  =",AVGDATA(1,15),"°F"
    PRINT #2, using form$: "EVAP REF Pg OUT =",AVGDATA(1,16),"psig"
    print #2, using form$: "Evap Ref dp     =",AVGDATA(1,14)-AVGDATA(1,16),"psid"
    PRINT #2, using form$: "COMP REF T  IN  =",AVGDATA(1,17),"°F"
    PRINT #2, using form$: "COMP REF Pg IN  =",AVGDATA(1,18),"psig"
    PRINT #2, using form$: "COMP REF T OUT  =",AVGDATA(1,19),"°F"
    PRINT #2, using form$: "COMP REF Pg OUT =",AVGDATA(1,20),"psig"
    PRINT #2, using form$: "COMP SPEED      =",AVGDATA(1,21),"RPM"
    PRINT #2, using form$: "COMP TORQUE     =",AVGDATA(1,22),"in•lb"
    PRINT #2, using form$: "AMB AIR PRESS   =",PAMB/6.89478  ,"psia"
    PRINT #2, using form$: "AMB AIR T DB    =",TDB*1.8 + 32,"°F"
    PRINT #2, using form$: "AMB AIR T WB    =",TWB*1.8 + 32,"°F"
    PRINT #2, using form$: "COND AIR DELT H =",(CAHO-CAHI)*0.42992,"Btu/lb"
    PRINT #2, using form$: "COND AIR MFLOW  =",CAFLO/.4536,"lb/hr"
    PRINT #2, using form$: "COND REF H IN   =",CRHI*0.42992,"Btu/lb"
    PRINT #2, using form$: "COND REF H OUT  =",CRHO*0.42992,"Btu/lb"
    PRINT #2, using form$: "EVAP AIR DELT H =",(EAHI-EAHO)*0.42992,"Btu/lb"
    PRINT #2, using form$: "EVAP AIR MFLOW  =",EAFLO/.4536,"lb/hr"
    PRINT #2, using form$: "EVAP REF H IN   =",ERHI*0.42992,"Btu/lb"
    PRINT #2, using form$: "EVAP REF H OUT  =",ERHO*0.42992,"Btu/lb"
    PRINT #2, using form$: "COMP REF H IN   =",KRHI*0.42992,"Btu/lb"
    PRINT #2, using form$: "COMP REF H OUT  =",KRHO*0.42992,"Btu/lb"
    PRINT #2, using form$: " "
    PRINT #2, using form$: "SUCT FLOW RATE  =",Msuc,"lb/hr"
    print #2, using form$: "Disch. Flow Rate =",Mdis,"lb/hr"
    print #2, using form$: "Liq. Flow Rate  =",Mliq,"lb/hr"
    PRINT #2, using form$: "CALCULATED FLOW =",CRFLOW/.4536,"lb/hr"
    PRINT #2: " "
    PRINT #2: " "
    print #2, using form$: "COP        =",Capacity/Power
    print #2, using form$: "Power      =",Power,"Btu/hr"
    print #2, using form$: "Capacity   =",Capacity,"Btu/hr"
    print #2, using form$: "Qcomp      =",Qcomp,"Btu/hr"
    print #2, using form$: "Qcond      =",Qcond,"Btu/hr"
    print #2, using form$: "Qsuction   =",Qsuct,"Btu/hr"
    print #2, using form$: "Qdischarge =",Qdisc,"Btu/hr"
    print #2: " "
    print #2, using form$: "Evap. Superheat  =",Esuper*1.8,"°F"
    print #2, using form$: "Cond. Subcooling =",Csub*1.8,"°F"
    print #2: " "
    print #2, using form$: "Condensation Rate =",dwater/.4536,"lb/hr"
    print #2: " "
    print #2: " "
  else

! SAVE AS COMMA-DELIMITD TEXT

    ask #3: filesize fs
    set #3: margin 600
    reset #3: end
    if 1=1 and fs=0 then
      print #3:sets
      print #3:prev
      if series$="s" then print #3:"Time  (sec),";
```

```
        print #3:"Con a Tin,Con a To,Con a flow,Con r Tin,Con r Pin,Con r To,Con r
                        Po,Con r ΔP,";
        print #3:"Evp a Tin,Evp a RHi,Evp a To,Evp a RHo,Evp a flow,Evp r Tin,Evp r
                        Pin,Evp r To,Evp r Po,";
        print #3:"Evp r ΔP,Com r Tin,Com r Pin,Com r To,Com r Po,Com Speed,Com
                        Torque,Amb air P,";
        print #3:"Amb air Tdb,Amb air Twb,Con a Δh,Con a Mdot,Con r hin,Con r ho,Evp
                        a Δh,Evp a Mdot,";
        print #3:"Evp r hin,Evp r ho,Com r hin,Com r ho,Suct Flow,Disc Flow,Liq
                        Flow,Calc Flow,COP,";
        print #3:"Power,Capacity,Qcomp,Qcond,Qsuction,Qdischarge,
                        Superheat,Subcooling,Condensation,Tcsat";
        print #3:",Tcondout"
      end if
      if series$="s" then print #3:tme;",";
      for i=1 to 7
        print #3:round(avgdata(1,i),2);",";
      next i
      print #3:round(avgdata(1,5)-avgdata(1,7),2);",";
      for i=8 to 16
        print #3:round(avgdata(1,i),2);",";
      next i
      print #3:round(avgdata(1,14)-avgdata(1,16),2);",";
      for i=17 to 22
        print #3:round(avgdata(1,i),2);",";
      next i
      print #3:round(PAMB/6.89478,2);",";round(TDB*1.8 + 32,2);",";
                        round(TWB*1.8 + 32,2);",";
      print #3:round((CAHO-CAHI)*0.42992,2);",";round(CAFLO/.4536,2);",";
                        round(CRHI*0.42992,2);",";
      print #3:round(CRHO*0.42992,2);",";round((EAHI-EAHO)*0.42992,2);",";
                        round(EAFLO/.4536,2);",";
      print #3:round(ERHI*0.42992,2);",";round(ERHO*0.42992,2);",";
                        round(KRHI*0.42992,2);",";
      print #3:round(KRHO*0.42992,2);",";round(Msuc,2);",";round(Mdis,2);",";
                        round(Mliq,2);",";
      print #3:round(CRFLOW/.4536,2);",";round(Capacity/Power,2);",";
                        round(Power,2);",";
      print #3:round(Capacity,2);",";round(Qcomp,2);",";round(Qcond,2);",";
                        round(Qsuct,2);",";
      print #3:round(Qdisc,2);",";round(Esuper*1.8,2);",";round(Csub*1.8,2);",";
                        round(dwater/.4536,3);
      print #3:",";round((Tcsat-273)*1.8+32,2);",";round((CRTO-273)*1.8+32,2)
    end if
  end if
  let tme=tme+intr
  if XXX=1 then
    let t2=time
    print tab(5,5); "Minutes remaining: "
  end if
  if XXX=1 or XXX/10=int(XXX/10) then
    print tab(5,24); "       "
    print tab(5,24);round((sets-XXX)*(t2-t1)/60,2)
  end if
```

```
NEXT XXX


! CLOSE ALL FILES

CLOSE #1
CLOSE #2
CLOSE #3

END


 sub wamb (PAMB,TDB,TWB,WIN)

 ! Calculate W for air given Pressure, Twb, and Tdb.
 ! Temperatures in C.  Pressure in kPa.

 let c1  = -5.6745359e3
 let c2  = -5.1523057e-1
 let c3  = -9.6778430e-3
 let c4  = 6.2215701e-7
 let c5  = 2.0747825e-9
 let c6  = -9.484024e-13
 let c7  = 4.1635019
 let c8  = -5.8002206e3
 let c9  = -5.5162560
 let c10 = -4.8640239e-2
 let c11 = 4.1764768e-5
 let c12 = -1.4452093e-8
 let c13 = 6.5459673

 let twb = twb + 273.15

 if twb < 273.15 then
   let pws = exp(c1/twb + c2 + c3*twb + c4*twb^2 + c5*twb^3 + c6*twb^4 +
c7*log(twb))
 else
   let pws = exp(c8/twb + c9 + c10*twb + c11*twb^2 + c12*twb^3 + c13*log(twb))
 end if

 let ws  = .62198*pws/(pamb-pws)
 let twb = twb - 273.15

 let win = ((2501-2.381*twb)*ws-(tdb-twb))/(2501+(1.805*tdb)-(4.186*twb))

 end sub


 sub moisth(p,phi,t,h,w)
 !
 ! MOIST AIR ENTHALPY GIVEN RELATIVE HUMIDITY AND TEMPERATURE.
 ! W IS HUMIDITY RATIO AT SPECIFIED RH AND TEMPERATURE
```

```
  !
  ! PRESSURE IN kPa
  ! TEMPERATURE IN K
  ! ENTHALPY IN KJ/KG

  let c1  = -5.6745359e3
  let c2  = -5.1523057e-1
  let c3  = -9.6778430e-3
  let c4  = 6.2215701e-7
  let c5  = 2.0747825e-9
  let c6  = -9.484024e-13
  let c7  = 4.1635019
  let c8  = -5.8002206e3
  let c9  = -5.5162560
  let c10 = -4.8640239e-2
  let c11 = 4.1764768e-5
  let c12 = -1.4452093e-8
  let c13 = 6.5459673

  if t < 273.15 then
    let pws = exp(c1/t + c2 + c3*t + c4*t^2 + c5*t^3 + c6*t^4 + c7*log(t))
  else
    let pws = exp(c8/t + c9 + c10*t + c11*t^2 + c12*t^3 + c13*log(t))
  end if

  let pw = phi/100.0 * pws
  let w  = 0.62198*pw/(p - pw)
  let h  = 1.005*(t-273.15) + w*(2501 + 1.805*(t-273.15))

  end sub


 Def Densliq(T)

   !TEMPERATURE (K), DENSITY (KG/M^3) DATA FOR R12

   !COEFFICIENTS...T=220-370 °K
   let C1=766.953
   let C2=-787.526
   let C3=5636.57
   let C4=-6671.89
   let C5=3202.81

   LET X = (1 - T/384.95)^(1/3)
   let Densliq = C1 + C2*X + C3*X^2 + C4*X^3 + C5*X^4
 end Def


 Def Hliq(T)

  !TEMPERATURE (K), LIQ. ENTHALPY (KJ/KG) DATA FOR R12
```

```
  !COEFFICIENTS...T= 220-370 °K
  let C1=624.164
  let C2=-5.53819
  let C3=.034685
  let C4=-8.42386e-5
  let C5=7.83359e-8

  let Hliq = C1 + C2*T + C3*T^2 + C4*T^3 + C5*T^4

end Def


Def Hvap(T)

  !TEMPERATURE (K), VAP. ENTHALPY (KJ/KG) DATA FOR R12

  If T>295 then
   !COEFFICIENTS...T= 270-370 °K
   let C1=-2156.97
   let C2=35.0819
   let C3=-.173056
   let C4=3.85794e-4
   let C5=-3.243e-7

   let Hvap = C1 + C2*T + C3*T^2 + C4*T^3 + C5*T^4
  Else
   !COEFFICIENTS...T=220-320 °K
   let C1=384.619
   let C2=1.53814
   let C3=-7.39091e-3
   let C4=2.2906e-5
   let C5=-2.68065e-8

   let hvap = C1 + C2*T + C3*T^2 + C4*T^3 + C5*T^4
  end if
end Def


Def P(T)

  !TEMPERATURE (K), PRESSURE (KPA) DATA FOR R12

  !COEFFICIENTS...T=220-370 °K
  let C1=-34.5426
  let C2=.371808
  let C3=-1.34106e-3
  let C4=2.33422e-6
  let C5=-1.58017e-9

  let lnp = C1 + C2*T + C3*T^2 + C4*T^3 + C5*T^4
  let P=exp(lnp)
end Def
```

```
Def Sliq(T)

 !TEMPERATURE (K), LIQ. ENTROPY (KJ/KG) DATA FOR R12

 if T>305 then
  !COEFFICIENTS...T=290-370 °K
  let C1=12.7429
  let C2=-.118725
  let C3=5.88984e-4
  let C4=-1.26023e-6
  let C5=1.00816e-9

  let Sliq = C1 + C2*T + C3*T^2 + C4*T^3 + C5*T^4
 else
  !COEFFICIENTS...T=220-320 °K
  let C1=2.66171
  let C2=9.06035e-3
  let C3=-1.75857e-5
  let C4=1.76573e-8

  let Sliq = C1 + C2*T + C3*T^2 + C4*T^3
 end if
end Def


Def Svap(T)

 !TEMPERATURE (K), VAP. ENTROPY (KJ/KG-K) DATA FOR R12

 if T<305 then
  !COEFFICIENTS...T=220-320 °K
  let C1=6.32369
  let C2=-1.49816e-2
  let C3=4.77448e-5
  let C4=-5.15929e-8

  let Svap = C1 + C2*T + C3*T^2 + C4*T^3
 else
  !COEFFICIENTS...T=290-370 °K
  let C1=-7.36119
  let C2=.15774
  let C3=-7.68432e-4
  let C4=1.65969e-6
  let C5=-1.34324e-9

  let Svap = C1 + C2*T + C3*T^2 + C4*T^3 + C5*T^4
 end if
end Def
```

```
Def T(P)

  !TEMPERATURE (K), PRESSURE (KPA) DATA FOR R12

  !COEFFICIENTS...P= 33.11-3154.1 kPa
  let C1=149.276
  let C2=26.1907
  let C3=-3.14803
  let C4=.409312

  let T = C1 + C2*log(P) + C3*(log(P))^2 + C4*(log(P))^3

End def


Def Vol(T)

  !TEMPERATURE (K), SPECIFIC VOLUME (M^3/KG) DATA FOR R12

  !COEFFICIENTS...T=220-370 °K
  let C1=13.7356
  let C2=7.65959
  let C3=-35.6923
  let C4=-32.4574
  let C5=8.36175

  LET TRATIO = T/384.95
  let lnv = C1 + C2/TRATIO + C3*TRATIO + C4*(1 - TRATIO)^(1.5) + C5*TRATIO^3
  let Vol=exp(lnv)
end Def

Def CPliq(T)

  !Liquid specific heat  T=170-370 °K

  let m0 =      -17.462567872
  let m1 =      0.42877339342
  let m2 =      -0.0041129649668
  let m3 =      2.0551213662e-05
  let m4 =      -5.6012650636e-08
  let m5 =      7.8277553688e-11
  let m6 =      -4.3101928075e-14

  let CPliq = m0 + m1*T + m2*T^2 + m3*T^3 + m4*T^4 + m5*T^5 + m6*T^6

end Def


Def Cvsup(T)

  !R12 CV superheat, T=270-360 °K  (EES)
```

```
  let m0 = 0.42314505035
  let m1 = 0.00024076022885
  let m2 = 6.8181391918e-07

  let CVsup = m0 + m1*T + m2*T^2

end Def


Def CPsup(T)

  !R12 CP superheat, T=270-360 °K  (EES)

  let m0 = 0.59530786818
  let m1 = -0.00029628042019
  let m2 = 1.4015038808e-06
  let CPsup = m0 + m1*T + m2*T^2

end Def


def rhoSHV(Tsuper,Tsat)
  dim c(16),f(1,16),temp(1),x1(1),x2(1)

  !coefficients...Tsat=224-306 °K

  let C(1)  = -390.06
  let C(2)  = 5.93507
  let C(3)  = -7.85603e-2
  let C(4)  = 4.81705e-4
  let C(5)  = 5.28265
  let C(6)  = -.07476
  let C(7)  = 9.68992e-4
  let C(8)  = -5.89571e-6
  let C(9)  = -2.43388e-2
  let C(10) = 3.16394e-4
  let C(11) = -3.99123e-6
  let C(12) = 2.40444e-8
  let C(13) = 3.83185e-5
  let C(14) = -4.51724e-7
  let C(15) = 5.50131e-9
  let C(16) = -3.27168e-11

  let x1(1)=Tsuper
  let x2(1)=Tsat

  DEF f1(x,z)=1
  DEF f2(x,z)=X
  DEF f3(x,z)=X^2
  DEF f4(x,z)=X^3
  DEF f5(x,z)=z
```

```
  DEF f6(x,z)=Z*X
  DEF f7(x,z)=Z*X^2
  DEF f8(x,z)=Z*X^3
  DEF f9(x,z)=Z^2
  DEF f10(x,z)=(Z^2)*X
  DEF f11(x,z)=(Z^2)*X^2
  DEF f12(x,z)=(Z^2)*X^3
  DEF f13(x,z)=Z^3
  DEF f14(x,z)=(Z^3)*X
  DEF f15(x,z)=(Z^3)*X^2
  DEF f16(x,z)=(Z^3)*X^3

  let i=1

  LET f(i,1)=f1(X1(i),X2(i))
  LET f(i,2)=f2(X1(i),X2(i))
  LET f(i,3)=f3(X1(i),X2(i))
  LET f(i,4)=f4(X1(i),X2(i))
  LET f(i,5)=f5(X1(i),X2(i))
  LET f(i,6)=f6(X1(i),X2(i))
  LET f(i,7)=f7(X1(i),X2(i))
  LET f(i,8)=f8(X1(i),X2(i))
  LET f(i,9)=f9(X1(i),X2(i))
  LET f(i,10)=f10(X1(i),X2(i))
  LET f(i,11)=f11(X1(i),X2(i))
  LET f(i,12)=f12(X1(i),X2(i))
  LET f(i,13)=f13(X1(i),X2(i))
  LET f(i,14)=f14(X1(i),X2(i))
  LET f(i,15)=f15(X1(i),X2(i))
  LET f(i,16)=f16(X1(i),X2(i))

  MAT temp=f*C
  let rhoSHV=temp(1)

END DEF



def hSHV(Tsuper,Tsat)
  dim c(16),f(1,16),temp(1),x1(1),x2(1)

  !coefficients...Tsat=224-306 °K

  let C(1)  = 679.864
  let C(2)  = -95.046
  let C(3)  = 5.27068
  let C(4)  = -5.88012e-2
  let C(5)  = -2.3407
  let C(6)  = 1.0926
  let C(7)  = -6.03105e-2
  let C(8)  = 6.72955e-4
  let C(9)  = 1.10903e-2
  let C(10) = -4.13218e-3
  let C(11) = 2.28069e-4
```

```
  let C(12) = -2.54482e-6
  let C(13) = -1.45752e-5
  let C(14) = 5.17965e-6
  let C(15) = -2.85292e-7
  let C(16) = 3.18285e-9

  let x1(1)=Tsuper
  let x2(1)=Tsat

  DEF f1(x,z)=1
  DEF f2(x,z)=X
  DEF f3(x,z)=X^2
  DEF f4(x,z)=X^3
  DEF f5(x,z)=z
  DEF f6(x,z)=Z*X
  DEF f7(x,z)=Z*X^2
  DEF f8(x,z)=Z*X^3
  DEF f9(x,z)=Z^2
  DEF f10(x,z)=(Z^2)*X
  DEF f11(x,z)=(Z^2)*X^2
  DEF f12(x,z)=(Z^2)*X^3
  DEF f13(x,z)=Z^3
  DEF f14(x,z)=(Z^3)*X
  DEF f15(x,z)=(Z^3)*X^2
  DEF f16(x,z)=(Z^3)*X^3

  let i=1

  LET f(i,1)=f1(X1(i),X2(i))
  LET f(i,2)=f2(X1(i),X2(i))
  LET f(i,3)=f3(X1(i),X2(i))
  LET f(i,4)=f4(X1(i),X2(i))
  LET f(i,5)=f5(X1(i),X2(i))
  LET f(i,6)=f6(X1(i),X2(i))
  LET f(i,7)=f7(X1(i),X2(i))
  LET f(i,8)=f8(X1(i),X2(i))
  LET f(i,9)=f9(X1(i),X2(i))
  LET f(i,10)=f10(X1(i),X2(i))
  LET f(i,11)=f11(X1(i),X2(i))
  LET f(i,12)=f12(X1(i),X2(i))
  LET f(i,13)=f13(X1(i),X2(i))
  LET f(i,14)=f14(X1(i),X2(i))
  LET f(i,15)=f15(X1(i),X2(i))
  LET f(i,16)=f16(X1(i),X2(i))

  MAT temp=f*C
  let hSHV=temp(1)

END DEF


def sSHV(T,Tsat)
```

```
!T=223-430 °K

dim c(25),f(1,25),yc(1),x1(1),x2(1)

let C(1) = -2039.09
let C(2) = 27.8276
let C(3) = -.14134
let C(4) = 3.17452e-4
let C(5) = -2.66073e-7
let C(6) = 85506.6
let C(7) = -1161.8
let C(8) = 5.88891
let C(9) = -1.31998e-2
let C(10) = 1.10412e-5
let C(11) = -1.05362e+6
let C(12) = 14283.9
let C(13) = -72.2359
let C(14) = .161532
let C(15) = -1.3479e-4
let C(16) = 4.94368e+6
let C(17) = -66819.
let C(18) = 336.841
let C(19) = -.750746
let C(20) = 6.24317e-4
let C(21) = -7.75655e+6
let C(22) = 104371.
let C(23) = -523.659
let C(24) = 1.1613
let C(25) = -9.60695e-4

DEF f1(x,z)=1
DEF f2(x,z)=X
DEF f3(x,z)=X^2
DEF f4(x,z)=X^3
DEF f5(x,z)=X^4
DEF f6(x,z)=log(z/x)
DEF f7(x,z)=log(z/x)*X
DEF f8(x,z)=log(z/x)*X^2
DEF f9(x,z)=log(z/x)*X^3
DEF f10(x,z)=log(z/x)*X^4
DEF f11(x,z)=(log(z/x))^2
DEF f12(x,z)=((log(z/x))^2)*X
DEF f13(x,z)=((log(z/x))^2)*X^2
DEF f14(x,z)=((log(z/x))^2)*X^3
DEF f15(x,z)=((log(z/x))^2)*X^4
DEF f16(x,z)=(log(z/x))^3
DEF f17(x,z)=((log(z/x))^3)*X
DEF f18(x,z)=((log(z/x))^3)*X^2
DEF f19(x,z)=((log(z/x))^3)*X^3
DEF f20(x,z)=((log(z/x))^3)*X^4
DEF f21(x,z)=(log(z/x))^4
DEF f22(x,z)=((log(z/x))^4)*X
DEF f23(x,z)=((log(z/x))^4)*X^2
DEF f24(x,z)=((log(z/x))^4)*X^3
```

```
  DEF f25(x,z)=((log(z/x))^4)*X^4

  let i=1
  let x1(1)=T
  let x2(1)=Tsat

  LET f(i,1)=f1(X1(i),X2(i))
  LET f(i,2)=f2(X1(i),X2(i))
  LET f(i,3)=f3(X1(i),X2(i))
  LET f(i,4)=f4(X1(i),X2(i))
  LET f(i,5)=f5(X1(i),X2(i))
  LET f(i,6)=f6(X1(i),X2(i))
  LET f(i,7)=f7(X1(i),X2(i))
  LET f(i,8)=f8(X1(i),X2(i))
  LET f(i,9)=f9(X1(i),X2(i))
  LET f(i,10)=f10(X1(i),X2(i))
  LET f(i,11)=f11(X1(i),X2(i))
  LET f(i,12)=f12(X1(i),X2(i))
  LET f(i,13)=f13(X1(i),X2(i))
  LET f(i,14)=f14(X1(i),X2(i))
  LET f(i,15)=f15(X1(i),X2(i))
  LET f(i,16)=f16(X1(i),X2(i))
  LET f(i,17)=f17(X1(i),X2(i))
  LET f(i,18)=f18(X1(i),X2(i))
  LET f(i,19)=f19(X1(i),X2(i))
  LET f(i,20)=f20(X1(i),X2(i))
  LET f(i,21)=f21(X1(i),X2(i))
  LET f(i,22)=f22(X1(i),X2(i))
  LET f(i,23)=f23(X1(i),X2(i))
  LET f(i,24)=f24(X1(i),X2(i))
  LET f(i,25)=f25(X1(i),X2(i))

  MAT yc=f*C
  let sSHV=yc(1)

end def


Def Mdvap(P,Td,dp)

  !Suction mass flow rate (lb/hr)

  Declare def rhoSHV,T

  let Tsat=T(P)
  let Tsuper=Td-Tsat
  let dens=rhoshv(Tsuper,Tsat)*.062428          !lb/ft^3
  if dp<-.015 then
    let Mdvap=-(48.487+416.13*sqr((dp+.015)*dens))
  else
    let Mdvap=48.487+416.13*sqr((dp+.015)*dens)
  end if
```

```
end Def


Def Mddisch(P,Td,dp)

   !Discharge mass flow rate (lb/hr)

   Declare def rhoSHV,T

   let Tsat=T(P)
   let Tsuper=Td-Tsat
   let dens=rhoshv(Tsuper,Tsat)*.062428          !lb/ft^3
   if dp<-.04 then
     let Mddisch=-(-70.561+280.78*sqr((dp+.04)*dens))
   else
     let Mddisch=-70.561+280.78*sqr((dp+.04)*dens)
   end if

end Def


Def Mdliq(T,dp)

   !Liquid mass flow rate (lb/hr)

   declare def Densliq
   let dens=Densliq(T)
   let dens=dens*.062428                    !lb/ft^3
   if dp<-0.007 then
     let Mdliq=-67.0537*sqr((abs(dp)+.007)*dens)
   else
     let Mdliq=67.0537*sqr((dp+.007)*dens)
   end if

end Def
```